



Complexite en moyenne de l'algorithme de multi-pattern matching RETE sur des ensembles de patterns et d'objets de profondeur un

Luc Albert

► To cite this version:

Luc Albert. Complexite en moyenne de l'algorithme de multi-pattern matching RETE sur des ensembles de patterns et d'objets de profondeur un. [Rapport de recherche] RR-1009, INRIA. 1989. inria-00075549

HAL Id: inria-00075549

<https://hal.inria.fr/inria-00075549>

Submitted on 24 May 2006

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



UNITÉ DE RECHERCHE
IRIA-ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
BP 105
78153 Le Chesnay Cedex
France
Tél. (1) 39 63 55 11

Rapports de Recherche

N° 1009

Programme 1

COMPLEXITE EN MOYENNE DE L'ALGORITHME DE MULTI-PATTERN MATCHING RETE SUR DES ENSEMBLES DE PATTERNS ET D'OBJETS DE PROFONDEUR UN

Luc ALBERT

Avril 1989



250

Complexité en moyenne de l'algorithme de multi-pattern matching RETE sur des ensembles de patterns et d'objets de profondeur un

*Luc ALBERT **

Résumé. L'algorithme RETE ([Forg 82]) est un algorithme très efficace pour effectuer le pattern-matching (ou semi-unification) d'un grand nombre d'objets avec un grand nombre de motifs (ou patterns), notamment dans les moteurs d'inférences des systèmes experts à Base de Règles. Nous avons étudié ([AF 88] ou [Alb 88]) la complexité en moyenne de l'algorithme RETE sur des ensembles de motifs et d'objets de structure arborescente quelconque. Objets et motifs sont souvent du type : un symbole de tête suivi d'une liste d'arguments constants ou variables (OPSV [Forg 81], Xrete [LCR 88] ...). Nous analysons, dans cet article, sa performance théorique pour ce type répandu de motifs et d'objets à l'aide de la théorie des séries génératrices. Nous étendons ces travaux à l'étude de la performance des requêtes multiples dans les Bases de Données Relationnelles et généralisons le théorème de Rosenthal sur la taille moyenne d'une équijointure [Rosen 81]. Nous développons aussi numériquement sur des exemples les résultats simples et précis obtenus.

Average case complexity analysis of RETE pattern-match algorithm on a collection of flat patterns and objects

Abstract. The RETE algorithm [Forg 82] is a very efficient method for comparing a large collection of patterns to a large collection of objects. It is widely used in rule-based expert systems. We studied ([AF 88] or [Alb 88]) the average case complexity of the RETE algorithm on collections of patterns and objects with a random tree structure. Objects and patterns are often made up of a head-symbol and of a list of variable or constant arguments (OPSV [Forg 81], Xrete [LCR 88] ...). In this paper, we analyse its theoretical performance on this widely used type of pattern and object with the theory of generating functions. We extend this work to the study of the performance of composed queries in relational Databases and we generalize Rosenthal's theorem on the average size of an equijoin [Rosen 81]. We apply numerically our simple and precise results to examples.

* Institut National de Recherche en Informatique et Automatique, Domaine de Voluceau, Rocquencourt, BP 105, 78150 Le Chesnay Cedex France,

Laboratoire Central de Recherches, Thomson-CSF, Domaine de Corbeville, BP 10, 91401 Orsay Cedex France.

I. Introduction

L'algorithme de multi-pattern matching **RETE** ([Forg 79], [Forg 82]) a été introduit par C. Forgy à la suite de ses travaux sur les systèmes de règles de production [Forg 81].

Les systèmes de règles de production sont très utilisés en Intelligence Artificielle pour modéliser un comportement intelligent [LNR 87] et réaliser des systèmes experts. Ils sont d'une utilisation assez souple et possèdent de nombreux avantages: modularité, indépendance relative des règles et grande puissance expressive (ils possèdent la généralité d'une machine de Turing). Toutefois, en contrepartie, le principal inconvénient réside dans l'inefficacité algorithmique des Moteurs d'Inférences. La phase la plus coûteuse dans un moteur d'inférence est la phase de pattern matching qui consiste à maintenir l'ensemble de toutes les instanciations des règles satisfaites, ou ensemble de conflit, durant les changements de la Base de Faits. Le système peut consacrer plus de 90 % de son temps à cette opération [DNM 78].

L'algorithme **RETE** détermine de façon incrémentale l'ensemble de conflit. En effet, pour les applications des systèmes experts le déclenchement d'une règle affecte un petit nombre de faits en comparaison avec le nombre de faits total de la Base de Faits. Cette détermination est donc efficace car il n'y a pas à chaque cycle du moteur d'inférence d'itération sur l'ensemble de la Base de Faits. D'autre part les conditions communes à plusieurs règles, qui sont fréquentes, sont partagées de façon à ce qu'en testant seulement quelques conditions on puisse déterminer la satisfaisabilité de plusieurs règles.

Forgy a montré, sous certaines hypothèses simplificatrices, dans [Forg 79] que la complexité dans le cas le pire pour déterminer l'ensemble des règles instanciées de l'algorithme **RETE** est linéaire en fonction du nombre de règles de la Base de Règles et polynômiale en le nombre de faits de la Base de Faits (avec pour degré le nombre supposé constant de conditions dans une règle). Dans le meilleur des cas la complexité en temps est constante. Entre ces deux extrêmes la complexité dépend complètement des caractéristiques des règles. Nous avons étudié la complexité moyennée théorique de l'algorithme **RETE** lorsque celui-ci semi-unifie des motifs et des objets de structure arborescente quelconque ([AF 88], [Alb 88]). En pratique les objets et les motifs sont souvent sous la forme : un symbole de tête suivi d'une liste d'arguments variables ou constants. Leur structure arborescente est donc de *profondeur un* ou "*plate*". Nous analysons ici la performance moyenne de l'algorithme dans ce cas fréquent.

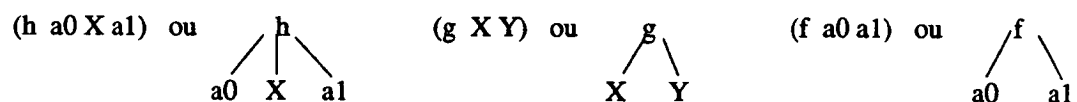


Figure 1 : Motifs plats et représentation arborescente

L'analyse en moyenne précise de l'algorithme **RETE** présente de nombreux intérêts. Tout d'abord, l'algorithme **RETE** admet beaucoup de variantes et d'optimisations: en ce qui concerne la représentation des mémoires locales [Forg 79], le calcul des jointures [Mir 87], la compilation totale du graphe [Fag 86] [Fag 88], la parallélisation de l'algorithme [Gupt 84], le partage des conditions (arbre d'unification : [Gha 87], [Alb 88]), etc ... On peut utiliser une analyse en moyenne de l'algorithme pour évaluer ces différentes optimisations et même en proposer de nouvelles. Il est nécessaire aussi de pouvoir prédire le temps d'exécution de l'algorithme pour le développement de systèmes experts temps réels [WGF 86], [SF 83]. Avec un modèle mathématique du temps d'exécution on peut extrapoler les performances d'un prototype aux performances d'un système expert grandeur nature et définir son domaine d'applicabilité en fonction du nombre d'objets susceptibles d'être traités à un instant donné. A l'aide d'un modèle mathématique peuvent être

aussi conçus des benchmarks significatifs pour comparer différentes implémentations en fonction des paramètres caractéristiques d'une base de connaissances [GF 83]. Enfin, comme nous le montrerons, cette analyse ne s'applique pas seulement à l'étude de la performance du pattern matching dans les systèmes experts, elle s'étend également à l'étude des requêtes multiples dans les Bases de Données Relationnelles. Dans ce papier nous présentons l'analyse en moyenne de l'algorithme **RETE** et celle de la taille moyenne d'une multi-équijointure dans les Base de Données Relationnelles à l'aide de la théorie des séries génératrices ([Fla 85], [FS 86], [FV 87]) (qui est intéressante en elle-même).

Comme nous l'avons dit, les objets ou faits (ou encore *termes*) sont représentés par un symbole de tête d'arité fixée et une liste d'arguments pris dans un ensemble de constantes, élargi à un ensemble { constantes + variables } pour les conditions (ou *motifs* en anglais *patterns*). Dans un premier temps on ne considérera que des tests d'égalité entre arguments. Dans la partie 2, on expose tout d'abord (2.1) succinctement l'algorithme **RETE**, notamment sur un exemple. Puis on définit la notion de coût de l'algorithme et les quantités fondamentales pour sa détermination (2.2). Ensuite dans la partie 3, on introduit la théorie des fonctions génératrices qui est employée pour analyser la complexité en moyenne des algorithmes (3.2) et l'analyse asymptotique afin de simplifier les résultats obtenus (3.3). On obtient alors un résultat avec un premier modèle où la Base de Faits est représentée par une liste ordonnée de faits (3.4). Dans la partie 4, on détermine la complexité en considérant la Base de Faits comme un multienemble de faits. Dans la partie 5, on étend les résultats obtenus en considérant plusieurs domaines distincts de variation pour les constantes (5.1). Puis on généralise de façon à introduire des pondérations traduisant des probabilités de présence différentes pour les symboles de tête et les constantes dans les différents domaines (5.2) (on envisagera même les tests d'inégalité). Dans la partie 6, on traite la négation entre arguments et entre motifs.

Enfin dans la partie 7, on montre comment tous ces résultats s'appliquent aisément à l'étude de la performance des requêtes multiples dans les bases de données relationnelles. Des exemples illustrent les résultats obtenus tout au long de l'article et l'exemple de la figure 2 est détaillé numériquement en Annexe.

II. L'algorithme RETE

Dans cette partie nous présentons brièvement l'algorithme **RETE** et quelques notations (2.1). Puis nous détaillons la notion de coût de l'algorithme (2.2).

2.1. Présentation

Les systèmes de règles de production que nous allons considérer par la suite sont composés d'un ensemble fixe, noté *BR* (pour *Base de Règles*), d'assertions du type "si-alors" appelées *règles de production*, et d'un ensemble variable de *faits* appelé *Base de Faits* ou *BF*. Les faits sont formés à partir d'un alphabet fini *F* de symboles de tête ou symboles fonctionnels donnés avec leur arité. Les arguments sont pris dans un ensemble fini *C* de symboles d'arité 0, les constantes. Par exemple, étant donné le symbole *h* d'arité 3, *f* d'arité 2 et les constantes *a1* et *a2* on peut former les termes suivants : (*f a1 a1*), (*f a2 a1*), (*h a1 a2 a1*), (*h a1 a2 a2*) etc ... On note l'ensemble des termes de profondeur 1 ou "plats" ainsi formés *TP(F)*. La Base de Faits est représentée par un ensemble ordonné ou non de tels termes.

La partie *-si-* d'une règle (son *membre gauche*) est une conjonction de conditions (ou *motifs*) représentée par le uplet (P_1, \dots, P_n) . Un motif est un terme dont certains arguments peuvent

être des variables. Les variables forment un ensemble dénombrable V et sont notées par X, Y, \dots . L'ensemble des motifs formés à partir de F et V est noté $TP(F, V)$. Les conditions filtrent les faits (*termes*). Plus précisément un terme t s'unifie avec un motif P si l'on peut trouver une *substitution* sur les variables du motif, $\sigma : V \rightarrow C$, telle que $\sigma P = t$. Par exemple le motif $(h \ X \ Y \ X)$, dans lequel on substitue X par $a1$ et Y par $a2$, s'unifie avec le terme $(f \ a1 \ a2 \ a1)$.

On dit que le membre gauche d'une règle (P_1, \dots, P_n) est *instancié* (ou que la règle est *satisfaite*), s'il existe un uplet de termes (t_1, \dots, t_n) avec $t_i \in BF$, appelé *l'instance*, et une substitution σ telle que $\sigma P_i = t_i$. Remarquons qu'un motif dans une règle pouvant filtrer plusieurs faits de la Base de Faits, une règle peut être instanciée de multiples façons. La partie *—alors—* d'une règle (son *membre droit*) est une suite d'actions qui peuvent avoir comme effet d'ajouter ou de supprimer un (ou plusieurs) fait de la Base de Faits.

Exemple : soient les symboles fonctionnels h d'arité 3, f et g d'arité 2 et les constantes $a0, a1, a2, a3, a4, a5$. On peut considérer les trois règles suivantes, où l'on a omis les membres droits :

$$\begin{aligned} (R1 : & (g \ X \ Y) \neg(f \ a0 \ X) (h \ a1 \ X \ Y) \longrightarrow \dots) \\ (R2 : & (h \ a0 \ X \ a1) (g \ X \ Y) (f \ a0 \ a1) \longrightarrow \dots) \\ (R3 : & (f \ X \ a1) (f \ Y \ a2) (h \ a2 \ X \ Y) \longrightarrow \dots) \end{aligned}$$

(La négation " \neg " est étudiée au 6.)

Les motifs-condition de la règle $R2$ sont ceux de la figure 1.

Le cycle d'inférences exécute les trois opérations suivantes :

- il *unifie* les motifs avec les faits de la Base de Faits afin de déterminer l'ensemble des règles instanciées (*l'ensemble de conflit*) : c'est la phase de *pattern-matching*.
- il *sélectionne* une instance particulière d'une règle dans l'ensemble de conflit ;
- il *déclenche* les actions de la règle choisie.

En fait, dans l'algorithme **RETE** la phase de déclenchement n'est pas séparée de la phase de pattern matching. Celle-ci est exécutée à chaque modification de la Base de Faits, c'est à dire à l'initialisation du système quand on entre les faits puis à chaque ajout ou suppression durant l'étape de déclenchement. Puisque l'algorithme de multi-pattern matching **RETE** s'intéresse uniquement aux membres gauches des règles, on identifiera à présent les règles avec leur membre gauche (on considérera les membres droits au 5.2 lorsque l'on distinguera des fréquences d'apparition des symboles de tête et des constantes). Les règles sont représentées sous la forme d'un graphe de discrimination : le **GRAPHE DE RETE**. Pendant l'exécution, quand un ajout ou une suppression est réalisé, un élément de modification parcourt le graphe depuis sa racine. Dès que celui-ci s'unifie avec un motif, il est mémorisé (ou enlevé) du graphe et si d'autres faits mémorisés s'unifient avec lui à un membre gauche de règle, l'instance est ajoutée (ou supprimée) de l'ensemble de conflit. En ce sens la Base de Règles et la Base de Faits sont toutes les deux représentées dans le graphe.

On peut distinguer deux types de tests dans le membre gauche des règles.

- Les tests *mono-motifs* sont les tests effectués au sein d'un motif pour le caractériser. Ils testent l'égalité des symboles de tête et des constantes entre un fait et le motif. Lorsqu'une même variable a plusieurs occurrences dans le motif, ils testent aussi l'égalité des constantes correspondantes du fait.
- Les tests *multi-motifs* testent différents motifs d'un même membre gauche de règle. Lorsqu'une même variable apparaît dans plusieurs motifs, ils vérifient que les constantes correspondantes sont bien égales. Ils effectuent une *jointure* au sens des Bases de Données (c'est à dire ces nœuds produisent en sortie un produit cartésien des données en entrée discriminées par des tests).

Dans l'algorithme **RETE** de multi-pattern matching, tout d'abord, on effectue les tests mono-motifs, ensuite on exécute les tests multi-motifs à chaque jointure sur des faits préalablement mémorisés. Le graphe de **RETE** se compose donc de deux parties. On a représenté en figure 2 le graphe de **RETE** correspondant aux trois règles précédentes.

La première partie est un arbre composé de tests mono-motifs, appelé *arbre de discrimination* ou *arbre test*. L'arbre de discrimination a autant de feuilles qu'il y a de motifs distincts dans les membres gauches des règles (l'égalité entre motifs étant entendue au renommage des variables près), soit 8 feuilles dans notre exemple. On définit le *i-motif* en un nœud i de l'arbre de discrimination comme le motif correspondant à la suite des tests effectués sur le chemin de l'arbre de discrimination allant de la racine à i compris, chemin que l'on appelle la *i-branche*. Le nœud racine est une position non testée, c'est à dire un motif non sélectif (qui s'unifie avec tous les faits), une variable. Les *i-motifs* des feuilles de l'arbre de discrimination sont tous les motifs distincts qui apparaissent dans tous les membres gauches des règles. Les différents successeurs d'un nœud correspondent aux différentes branches du graphe à parcourir. Lors de l'exécution un fait atteint les différentes feuilles qui correspondent aux différents motifs avec lesquels il s'unifie.

Ainsi dans notre exemple, au nœud 8 correspond le motif $(h \ a0 \ X \ Y)$ et le terme $(f \ a0 \ a1)$ parvient aux nœuds 4, 5 et 10.

La seconde partie du graphe de **RETE** est composée de jointures *binaires* entre les feuilles de l'arbre de discrimination. En ces nœuds-jointure, le *i-motif* est un uplet de motifs (P_1, \dots, P_l) , dans lequel la présence de variables communes inter-motifs est testée par les tests multi-motifs. Les nœuds terminaux de ce *graphe de jointure* correspondent bijectivement avec les règles de la Base de Règles. Les *i-motifs* des nœuds terminaux sont les membres gauches de toutes les règles au renommage des variables près. Ce sont donc les instances des règles qui atteignent les nœuds terminaux. Lors de l'exécution, lorsqu'un uplet parvient en un nœud-jointure, il est mémorisé dans une mémoire locale un *nœud-mémoire*, et l'on parcourt alors le nœud-mémoire de l'autre entrée du nœud-jointure afin de trouver un uplet satisfaisant aux tests multi-motifs. Ainsi au nœud 17 de la figure 2, en entrée gauche sont mémorisés des doublets du type $(f \ X \ a1)(f \ Y \ a2)$, en entrée droite sont mémorisés des $(h \ a2 \ U \ V)$ et on sélectionne des triplets suivants les tests multi-motifs $X = U$ et $Y = V$. Dans beaucoup d'implémentations on restreint souvent l'entrée droite d'un nœud-jointure à être directement une feuille de l'arbre de discrimination (et non une sortie d'un autre nœud-jointure) comme dans la figure 2 de sorte que le graphe de jointure est un ensemble de *peignes*. Ceci élimine une possibilité d'optimisation consistant à regrouper ensemble les motifs les plus "sélectifs" (nous nous placerons donc dans le cas général pour mener les calculs).

La complexité en temps de calcul en un nœud-jointure est quadratique en la taille de ses entrées. Dans le pire des cas la taille mémoire des feuilles de l'arbre de discrimination est proportionnelle à la taille $|BF|$ de la Base de Faits. Donc, dans le pire des cas, le calcul de l'ensemble de conflit est polynômial en $|BF|$, le degré étant le nombre maximum de motifs dans une règle.

On peut utiliser les techniques de hachage pour réduire la complexité de parcours des nœuds mémoire, en hachant les mémoires d'entrées suivant les tests d'égalités. (De même, si l'on considère des tests d'inégalité, on peut structurer les nœuds mémoires en arbres de recherche).

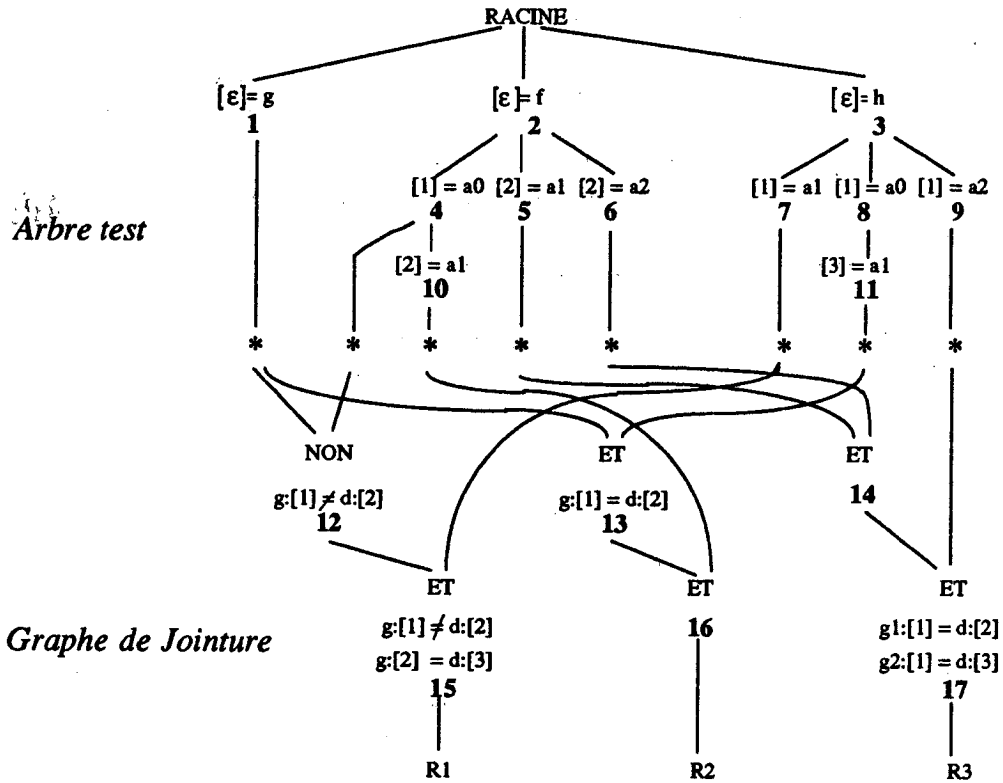


Figure 2 † : Graphe de Rete

Nous n'avons pas encore expliqué la négation de motifs. Un motif nié traduit la non-existence d'un fait s'unifiant avec le motif. Les motifs niés sont traités dans l'algorithme **RETE** par un second type de nœuds jointures (nœud 12). En ces nœuds le nœud-mémoire de l'entrée gauche conserve pour chaque uplet le nombre d'éléments de la mémoire droite (les éléments niés) qui lui sont compatibles relativement aux test multi-motifs. Quand un compteur atteint 0 (resp. 1) le uplet de gauche est propagé dans le graphe sous la forme d'un ajout (resp. d'une suppression). Dans notre exemple, tant qu'il n'y a pas dans la Base de Faits de termes du type $(f \text{ } ao \text{ } X)$, le terme en entrée du nœud 12 est transmis au nœud 15. Le quantificateur existentiel peut être traité de façon analogue. En fait il est possible de généraliser l'algorithme **RETE** de façon à accepter dans les membres gauches des règles toute formule logique du premier ordre avec un degré d'imbrication de quantificateurs quelconque ([Fag 88]).

2.2. Coût de l'algorithme

L'algorithme **RETE** passe la partie prépondérante de son temps dans la phase de pattern-matching. On va déterminer le coût moyen de l'algorithme **RETE**, étant donné une Base de Faits quelconque et un graphe de **RETE** fixé. La moyenne sera effectuée sur toutes les Bases de Faits probables. Plus

† Notation : on note par $[j]$ le $j^{ième}$ fils le plus à gauche de la racine ($[\epsilon]$) du i -motif ; d (g) désigne le motif en entrée droite (gauche) d'un nœud-jointure (g_j précise le $j^{ième}$ motif de la liste en entrée gauche).

précisément, nous allons évaluer le temps moyen nécessaire à l'algorithme **RETE** pour déterminer l'ensemble de conflit à partir d'une Base de Faits d'entrée donnée quelconque. Dans ce coût moyen, le temps de parcours des faits dans l'arbre de discrimination du graphe de **RETE** reste négligeable par rapport au temps consacré aux jointures. Nous

allons donc chercher à déterminer le temps moyen passé à réaliser les tests multi-motifs aux nœuds-jointure du graphe de **RETE**. Notre mesure de complexité sera donc le temps de réalisation d'un test multi-motif. En un nœud-jointure, le temps moyen total est proportionnel au nombre moyen de uplets de faits à tester en ce nœud. Lorsqu'il n'y a pas de hachage aux nœuds-jointure, cette quantité est le produit des nombres moyens de uplets de faits stockés aux nœuds-mémoire en entrée gauche et droite du nœud-jointure (le produit des *tailles* moyennes des nœuds-mémoire). Lorsqu'il y a hachage de ces mémoires locales suivant les tests d'égalité réalisés au nœud-jointure, le nombre moyen de uplets de faits à tester coïncide avec le nombre moyen de uplets de faits qui sortent avec succès du nœud-jointure considéré, soit la taille moyenne du nœud-mémoire de sortie du nœud-jointure.

Il apparaît donc comme quantité fondamentale à calculer, la taille moyenne des nœuds-mémoire du graphe de **RETE**. Nous pourrions donc ainsi déterminer le coût moyen de l'algorithme pour une Base de Règles donnée et l'encombrement moyen précis de chacune des mémoires locales du graphe.

Pour cela, nous pouvons modéliser la Base de Faits d'entrée comme *une liste ordonnée de faits*. Ceci représente bien la succession d'entrée dans le graphe de **RETE** des termes de modification et leur possible répétition au cours des cycles d'inférences (suppression puis ajout d'un même fait). Ceci va donner lieu aux calculs dans le cadre du modèle de liste. On peut également, puisque l'état final des mémoires locales du graphe de **RETE** est indépendant de l'ordre d'ajout des faits de la Base de Fait d'entrée, effectuer une moyenne sur *des ensembles de faits non ordonnés (autorisant les répétitions)*. Ceci donnera lieu aux calculs menés dans le cadre du modèle de multiensemble de termes.

III. Calculs dans le cadre du modèle de liste

Dans cette partie on va déterminer la *taille* moyenne des nœuds-mémoire du graphe de **RETE** en modélisant la Base de Faits par une liste ordonnée de faits. On présente tout d'abord (3.1) le modèle considéré puis on effectue les calculs combinatoires (3.2), dont on simplifie les résultats à l'aide d'une étude asymptotique (3.3). On en déduit enfin le coût moyen de l'algorithme (3.4).

3.1. Modèle de liste

On sait par [SF 83] et [Stey 84] que le coût moyen pour unifier un motif de taille p (i.e. avec une représentation arborescente de p nœuds) à un terme de taille n est constant. Dans le cas de l'algorithme **RETE** on considère des semi-unifications avec une conjonction de motifs.

A cette définition usuelle de la taille $|t|$ d'un terme t , on associe classiquement la série génératrice :

$$P(z) = \sum_{t \in TP(F)} z^{|t|} = \sum_{n \geq 0} p_n z^n$$

avec p_n le nombre de termes de taille n de $TP(F)$. Notons c_j le nombre de symboles de tête d'arité j , ainsi c_0 nous donne le nombre de constantes. Puisqu'il y a un nombre fini de symboles de tête, d'arité comprise entre m (pour l'arité minimum) et M (pour l'arité maximum), puisqu'il y a un

nombre fini de constantes c_0 , le nombre de termes constructibles de $TP(F)$ est lui aussi fini et $P(z)$ est donc un polynôme de degré $M + 1$. Plus précisément, on a :

$$P(z) = \sum_{l=m+1}^{M+1} (c_{l-1}c_0^{l-1})z^l$$

Nous allons introduire pour des commodités de calcul, une nouvelle définition de la taille adaptée à la structure des termes étudiés : le nombre des arguments moins m . Ainsi on introduit une nouvelle série génératrice :

$$A(z) = \sum_{i=0}^{M-m} a_i z^i \quad \text{avec} \quad a_i = c_{i+m} c_0^{i+m}$$

a_n étant le nombre de termes de $TP(F)$ de taille (i.e. de "nouvelle taille") n ; notre convention de taille assurant que $a_0 \neq 0$ ($A(z)$ est un polynôme de plus bas degré $M - m$). Dans la plupart des Bases de Faits, les termes sont de taille usuelle assez petite (4, 5, 6 ...). On peut introduire la moyenne des tailles usuelles (resp. nouvelles) ou taille moyenne usuelle (resp. nouvelle) des termes de $TP(F)$ notée $\frac{1}{\lambda_u}$ (resp. $\frac{1}{\lambda}$), qui vaut immédiatement :

$$\frac{1}{\lambda_u} = \frac{P'(1)}{P(1)} \quad \text{et} \quad \frac{1}{\lambda} = \frac{A'(1)}{A(1)}$$

avec bien sur $1/\lambda_u = 1/\lambda + m + 1$. Remarquons que $P(1) = A(1)$ représente le cardinal de $TP(F)$.

Remarque : Dans le cas où $m = M$, on a $\frac{1}{\lambda} = 0$!! Ce cas particulier sera donc traité à part et dorénavant on a $0 < \frac{1}{\lambda} < M - m$.

Exemple : dans notre exemple précédent, on a $c_0 = 6$, $c_1 = 0$, $c_2 = 2$ et $c_3 = 1$. On a donc $m = 2$. $M = 3$ et :

$$A(z) = 72 + 216z \quad \text{et} \quad P(z) = 72z^3 + 216z^4 = z^3 A(z)$$

On a $\frac{1}{\lambda} = 3/4$ et $\frac{1}{\lambda_u} = 15/4$.

Nous considérons en entrée du graphe de **RETE**, une Base de Faits de k termes de taille globale n (n est la somme des tailles des termes composant la Base de Faits). La série génératrice d'une liste de k termes de $TP(F)$ déterminée par $A(z)$ est donnée par $A(z)^k$. Ainsi le $n^{i\text{ème}}$ coefficient de z dans $A(z)^k$, $[z^n]A(z)^k$, représente le nombre de listes (ordonnées) de taille globale n à k éléments de $TP(F)$.

3.2. Formules combinatoires exactes

Pour un nœud i de l'arbre de discrimination, on pose : $B_i(z) = \sum_{n \geq 0} b_{i,n} z^n$, $b_{i,n}$ étant le nombre de termes de taille n filtrés (avec succès) en i .

Pour un nœud i du graphe de jointure qui produit en sortie des l -uplets de termes qui instancient partiellement le membre gauche d'une règle, on pose : $B_i(z) = \sum_{n \geq 0} b_{i,n} z^n$ avec $b_{i,n}$ le nombre de l -uplets de taille totale n qui sont filtrés en ce nœud-jointure i .

Introduisons quelques notations : on note pour un nœud i de l'arbre de discrimination

- ω_i l'arité du symbole de tête du i -motif,
- x_i le nombre d'arguments variables distincts du i -motif (en considérant que tout argument non précisé du symbole de tête est un nouvel argument variable).

Exemple : pour le nœud 8 du graphe de la figure 2, on a $\omega_8 = 3$ et $x_8 = 2$. Pour le motif (*head* $XXY a$), on aurait $\omega_i = 4$ et $x_i = 2$.

Il s'en suit immédiatement l'expression de $B_i(z)$:

THÉORÈME 1 : La série génératrice des termes filtrés en un nœud i de l'arbre de discrimination est :

$$B_i(z) = c_0^{x_i} z^{\omega_i - m}$$

PREUVE : les termes filtrant le i -motif sont en effet tous de même taille (déterminée par le symbole de tête) et chaque nouvelle feuille variable donne lieu à c_0 instanciations différentes. ■

Exemple : Pour le nœud 11 de la figure 2, on a $B_{11}(z) = 6z$.

Remarque : on a bien $B_{racine}(z) = A(z)$.

Considérons à présent un nœud i du graphe de jointure. i produit en sortie des listes de l -uplets. On connaît les l feuilles de l'arbre de discrimination qui ont filtré chacune une des composantes des l -uplets testés en i . Soit L_i cet ensemble de l motifs. Notons alors :

$$\begin{aligned} - \omega_i &= \sum_{j' \in L_i} \omega_{j'} \\ - x_i &= \left(\sum_{j' \in L_i} x_{j'} \right) + y_i \quad y_i \leq 0 \end{aligned}$$

avec y_i le terme correctif dû aux tests multi-motifs d'égalité entre variables des l différents motifs. Si l'on considère l'ensemble global de toutes les variables des l motifs, x_i compte encore le nombre de variables distinctes dans cet ensemble. On détermine donc aisément y_i en fonction des égalités entre variables testées jusqu'en i : on a d'ailleurs : $y_i \leq 0$ car les jointures accroissent *a priori* le nombre de variables identiques.

Ainsi pour le nœud 13 de la figure 1, où l'on teste les doublets $(h \text{ a0 } X \text{ a1})(g \text{ X } Y)$ en provenance des nœuds "associés" $j_1^{13} = 1$ et $j_2^{13} = 11$, on a $\omega_{13} = 5$, $x_{13} = 2$ et $y_{13} = -1$.

On obtient alors encore immédiatement :

THÉORÈME 2 : La série génératrice des l -uplets de termes qui sont filtrés au nœud i du graphe de jointure est :

$$B_i(z) = c_0^{x_i} z^{\omega_i - lm}$$

Déduisons de ceci le nombre moyen $\overline{b_{i,n,k}}$ de termes filtrés en i , nœud de l'arbre de discrimination, en considérant comme donnée de l'algorithme RETE une Base de Faits quelconque de k termes de taille n .

THÉORÈME 3 : Le nombre moyen de termes filtrés en un nœud i de l'arbre de discrimination est :

$$\overline{b_{i,n,k}} = \frac{k c_0^{x_i} [z^{n-\omega_i+m}] A(z)^{k-1}}{[z^n] A(z)^k}$$

PREUVE : l'idée de base de cette démonstration est de séparer $A(z)$ en : $A(z) = B_i(z) + R_i(z)$ avec $R_i(z)$ la "série reste" c'est à dire la série génératrice de tous les termes qui ne s'unifient pas avec le i -motif. Alors, on marque avec la variable u les éléments filtrés en i et on pose:

$$A(z, u) = u B_i(z) + R_i(z) = A(z) + (u - 1) B_i(z)$$

Ainsi $f_{i,n,p,k} = [u^p z^n] (A(z, u)^k)$ représente le nombre de listes de k termes de taille n dont exactement p termes sont filtrés en i .

Donc $\overline{b_{i,n,k}}$ est le quotient de $\sum_{p=0}^{+\infty} p f_{i,n,p,k} (= \sum_{p=0}^k p f_{i,n,p,k})$ par le nombre total de listes de taille globale n de k éléments, soit :

$$\overline{b_{i,n,k}} = \frac{\sum_p p f_{i,n,p,k}}{[z^n] A(z)^k}$$

De $(A(z, u)^k) = \sum_{p,n} f_{i,n,p,k} u^p z^n$ on tire :

$$\frac{\partial}{\partial u} (A(z, u)^k) = \sum_{p,n} f_{i,n,p,k} p u^{p-1} z^n$$

ou

$$\left. \frac{\partial}{\partial u} (A(z, u)^k) \right|_{u=1} = \sum_n \left(\sum_p p f_{i,n,p,k} \right) z^n$$

Donc

$$\overline{b_{i,n,k}} = \frac{[z^n] \frac{\partial}{\partial u} (A(z, u)^k) \Big|_{u=1}}{[z^n] A(z)^k}$$

Comme

$$\begin{aligned} \left. \frac{\partial}{\partial u} (A(z, u)^k) \right|_{u=1} &= k A(z, u)^{k-1} \frac{\partial}{\partial u} (A(z, u)) \Big|_{u=1} \\ &= k A(z)^{k-1} B_i(z) \end{aligned}$$

le théorème est démontré. ■

Nous allons à présent déterminer le nombre moyen de l -uplets de termes filtrant en un nœud i du graphe de jointure.

Considérons les l feuilles j^i de l'arbre de discrimination associées au nœud-jointure i (qui filtrent chacune, indépendamment de tout test multi-motifs, une des composantes des l -uplets testés en i). Une démonstration similaire à celle du théorème 3. du type séparation des termes de $A(z)$ en $A(z) = \sum_{j^i=1}^l B_{j^i}(z) + R(z)$ est impossible puisqu'il peut y avoir duplication des termes aux feuilles de l'arbre de discrimination (un même terme de la Base de Faits peut filtrer plusieurs motifs-condition des règles de la Base de Règles); d'autre part considérer $A(z)^l$ empêche toute moyenne sur une Base de Faits en entrée quelconque (Cf. [Alb 88] et [AF 88]). Nous allons donc aborder ce dénombrement de façon différente. Le calcul va refléter les deux aspects antagonistes, du point de vue du dénombrement, d'une jointure. En effet, on sait que la taille de la mémoire locale de sortie d'un nœud-jointure est conditionnée par deux phénomènes : l'*expansion* due au produit cartésien des données d'entrée du nœud et d'autre part la *diminution* due à la sélection suivant les tests multi-motifs de la jointure.

Pour une Base de Faits d'entrée de k termes et de taille globale n , on connaît, d'après le théorème 3, le nombre moyen de termes filtrés aux feuilles de l'arbre de discrimination. Pour un nœud-jointure i , on connaît également les l feuilles de l'arbre de discrimination (j^i) associées à i . On dispose donc des $\overline{b_{j^i,n,k}}$ et l'aspect "*expansion*" du nœud-jointure i peut s'exprimer par la quantité $\overline{L} = \prod_{j^i=1}^l \overline{b_{j^i,n,k}}$ qui représente (sous l'hypothèse de lois de présence des termes aux nœuds j^i indépendantes) le nombre moyen de l -uplets de termes "candidats" au filtrage en i (\overline{L} est exactement le nombre de l -uplets qui entreraient en i si tous les nœuds-jointure précédant i n'effectuaient que des produits cartésiens). Quant à l'aspect "*sélection*" au nœud i due à la jointure, on peut l'exprimer en calculant le nombre moyen de l -uplets de termes qui filtrent en i à partir de \overline{L} l -uplets de termes dont chacune des composantes filtre un des nœuds associés à i . On a :

THÉORÈME 4 : En un nœud i du graphe de jointure, le nombre moyen de l -uplets de termes filtrés en i à partir d'une liste de taille globale n de L l -uplets de termes dont chacune des composantes est filtré par un nœud j^i est :

$$\overline{\gamma_{i,L,n}} = L \frac{[z^n] B_i(z) \left(\prod_{j^i=1}^l B_{j^i}(z) \right)^{L-1}}{[z^n] \left(\prod_{j^i=1}^l B_{j^i}(z) \right)^L}$$

les j^i étant les l feuilles de l'arbre de discrimination associées à i .

PREUVE : La série génératrice des l -uplets dont chacune des composantes filtre une feuille de l'arbre de discrimination associée à i est :

$$D(z) = \prod_{j^i=1}^l B_{j^i}(z)$$

Si l'on marque par w les l -uplets qui filtrent jusqu'en i (avec les tests multi-motifs) on obtient :

$$D(z, w) = (w - 1) B_i(z) + \prod_{j^i=1}^l B_{j^i}(z)$$

D'où, si l'on considère une liste de L l -uplets, le nombre de telles L -listes de taille n qui comportent r l -uplets exactement qui filtrent en i est $[w^r z^n] D^L(z, w)$.

On a donc

$$\overline{\gamma_{i,L,n}} = \frac{\sum_{r=1}^L r [w^r z^n] \left((w - 1) B_i(z) + \prod_{j^i=1}^l B_{j^i}(z) \right)^L}{[z^n] \left(\prod_{j^i=1}^l B_{j^i}(z) \right)^L}$$

Et donc comme pour le théorème 3, on obtient bien l'expression de $\overline{\gamma_{i,L,n}}$. ■

On déduit du théorème 4 :

THÉORÈME 5 : Le nombre moyen de l -uplets de termes filtrant en un nœud i du graphe de jointure est :

$$\overline{b_{i,n,k}} = c_0^{y_i} \left(\prod_{j^i=1}^l \overline{b_{j^i,n,k}} \right) \quad (y_i \leq 0)$$

les nœuds j^i étant les feuilles de l'arbre de discrimination associées au nœud i .

PREUVE : en effet il suffit de remplacer dans le théorème 4, L par la valeur \overline{L} précédemment déterminée. On a dans ce cas : $n = \overline{L}(\omega_i - lm)$ et

$$\prod_{j^i=1}^l B_{j^i}(z) = c_0^{x_i - y_i} z^{\omega_i - lm} \quad \text{et} \quad B_i(z) = c_0^{x_i} z^{\omega_i - lm}$$

d'où $\overline{\gamma_{i,\overline{L},n}} = \overline{L} c_0^{y_i}$ d'où le résultat. ■

Nous obtenons donc des résultats en fonction des caractéristiques du graphe de RETE qui est une donnée fixe de l'algorithme. Ces résultats nous permettront donc une optimisation dans la

création du graphe de **RETE** (construction pour laquelle il y a encore beaucoup de choix arbitraires dans l'ordonnancement des nœuds).

La valeur combinatoire exacte $\overline{b_{i,n,k}}$ pour un nœud i de l'arbre de discrimination pourrait être étudiée en utilisant, par exemple, la formule d'inversion de Lagrange mais l'expression obtenue est difficile à utiliser (coefficients multinômiaux...). De façon à simplifier ces résultats nous allons effectuer une évaluation asymptotique en fonction de k et n , car on peut en effet considérer ces deux paramètres comme arbitrairement grands si l'on se réfère aux grosses Bases de Faits des systèmes réels, ou, ce qui revient au même, si l'on considère beaucoup de cycles du moteur d'inférence (et donc beaucoup d'ajouts et de suppressions de termes).

3.3. Evaluation asymptotique

Afin de simplifier les expressions obtenues, nous allons utiliser des méthodes d'analyse de singularités. Nous allons estimer la valeur d'intégrales avec des méthodes d'analyse complexe (comme la méthode du point col). On considère comme donnée de l'algorithme une Base de Faits de taille globale n avec k termes, avec n et k grands tous les deux.

D'après la section 3.2 nous devons déterminer des coefficients du type $[z^n]A(z)^k$, dont on déduira l'expression de $\overline{b_{i,n,k}}$.

Soit $a_{n,k} = [z^n]A(z)^k$. Par le théorème de Cauchy on peut exprimer cette quantité par l'intégrale suivante

$$a_{n,k} = \frac{1}{2i\pi} \int_{\mathcal{L}} (A(z)^k) \frac{dz}{z^{n+1}}$$

où \mathcal{L} est un lacet qui entoure simplement l'origine en restant dans le domaine d'analyticit  de $A(z)$. Afin d'obtenir un  quivalent de cette int grale quand k et n tendent tous deux vers $+\infty$, nous allons utiliser la m thode du col (cf. [dB 58] ou [Henr 74]). Afin de simplifier cette analyse asymptotique, nous allons profiter du fait que, dans les impl mentations r elles, la taille des termes est approximativement constante; ceci nous permet de poser $k = \lambda n$ avec $\frac{1}{\lambda}$ la taille moyenne des termes. Ainsi nous n'avons plus   consid rer que le seul param tre n tendant vers $+\infty$ (Pour les calculs qui vont suivre, on pourra se reporter   [Alb 87] ou   [Alb 88] o  des calculs analogues sont d velopp s). $a_{n,k}$ devient alors :

$$a_{n,k} = \frac{1}{2i\pi} \int_{\mathcal{L}} (A(z)^{\lambda n}) \frac{dz}{z^{n+1}}$$

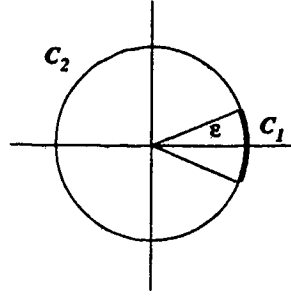
Nous allons utiliser la *m thode de col*. Cela va nous conduire   choisir un lacet \mathcal{L} particulier. Quand n cro t, la partie principale de l'int grale est $(A(z)^\lambda/z)^n$. Si on pose $g(z) = \ln(A(z)^\lambda/z)$, on peut choisir un point col   partir de l' quation fondamentale :

$$g'(z) = 0 = \lambda \frac{A'(z)}{A(z)} - \frac{1}{z} \quad \text{soit} \quad \lambda z A'(z) = A(z)$$

Or, on a $\lambda = A(1)/A'(1)$; donc 1 est racine de l' quation fondamentale; nous allons donc choisir comme contour d'int gration le cercle unit  \mathcal{C} . L'id e de base ensuite consiste   couper l'int grale en deux :

$$a_{n,k} = \int_{\mathcal{C}} = \int_{\mathcal{C}_1} + \int_{\mathcal{C}_2} = I_1 + I_2$$

avec $\mathcal{C}_1 = \{z/|z|=1 \text{ et } -\epsilon \leq \text{Arg}(z) \leq \epsilon\}$, $\mathcal{C}_2 = \mathcal{C}/\mathcal{C}_1$ et $\epsilon = \ln^2 n/\sqrt{n}$. On distingue ainsi la contribution de l'unit  et celle du reste du cercle dans I .



Ce choix de ϵ est tel que d'une part I_2 est exponentiellement négligeable devant I_1 et que d'autre part ϵ est assez petit pour que l'on puisse réaliser des développements sous le signe somme afin d'estimer la valeur de I_1 .

3.3.1 I_2 négligeable

On va montrer que I_2 est exponentiellement négligeable devant $A(1)^{\lambda n}$. On a :

$$I_2 = \frac{1}{2i\pi} \int_{C_2} (A(z)^{\lambda n}) \frac{dz}{z^{n+1}}$$

et grâce à notre convention de taille $a_0 \neq 0$. Il va falloir distinguer suivant la valeur de a_1 .

3.3.1.1 a_1 est non nul

Dans ce cas, on a

$$\begin{aligned} |A(z)| &= |A(e^{i\theta})| = \left| \sum_{k \geq 0} a_k e^{ik\theta} \right| \\ &\leq |a_0 + a_1 e^{i\theta}| + \left| \sum_{k \geq 2} a_k e^{ik\theta} \right| \\ &\leq |a_0 + a_1 e^{i\theta}| + \left| \sum_{k \geq 2} a_k \right| \\ &= |a_0 + a_1 e^{i\theta}| + A(1) - a_0 - a_1 \end{aligned}$$

On a $|a_0 + a_1 e^{i\theta}| \leq |a_0 + a_1 e^{i\epsilon}|$ (car $\cos(\theta) \leq \cos(\epsilon)$). Et en développant pour ϵ petit, on obtient $|a_0 + a_1 e^{i\epsilon}| = a_0 + a_1 - K\epsilon^2$ avec K une constante strictement positive. D'où, pour n assez grand, $\exists d > 0$ tel que :

$$|A(z)| \leq A(1)(1 - d\epsilon^2)$$

d'où

$$\begin{aligned} I_2 &\leq \frac{1}{2\pi} \int_{C_2} |A(1)|^{\lambda n} (1 - d\epsilon^2)^{\lambda n} d\theta \\ &\leq A(1)^{\lambda n} (1 - d\epsilon^2)^{\lambda n} \end{aligned}$$

A présent, puisque $\epsilon = \ln^2 n / \sqrt{n}$, on a

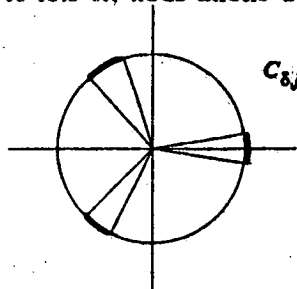
$$(1 - d\epsilon^2)^{\lambda n} = e^{\lambda d \ln^4 n + O(\frac{\ln^6 n}{n})} = O(n^{-\lambda d \ln^3 n})$$

et donc I_2 est bien exponentiellement négligeable devant $A(1)^{\lambda n}$.

Traisons maintenant le cas où $a_1 = 0$. Il faut considérer le *pgcd* des exposants de $A(z)$ (i.e. le *pgcd* des arités des symboles de tête $-m$). Nous allons devoir discuter suivant la valeur de ce *pgcd* ($= \delta$).

3.3.1.2 Le *pgcd* $\delta > 1$

C'est à dire $A(z) = N(z^\delta)$. Cette fois-ci, nous allons devoir sommer les contributions des racines δ -ièmes de l'unité.



I_2 va représenter cette fois-ci l'intégrale sur le contour

$$\mathcal{C}_2 = \mathcal{C} / \bigcup_{j=1}^{\delta} \mathcal{C}_{\delta,j} \quad \text{avec} \quad \mathcal{C}_{\delta,j} = \{e^{i\theta}, \theta \in [\frac{2j\pi}{\delta} - \epsilon, \frac{2j\pi}{\delta} + \epsilon]\}$$

Sur \mathcal{C}_2 , I_2 est toujours exponentiellement négligeable car $\cos(\delta\theta) \leq \cos(\delta\epsilon)$ et on a encore la majoration

$$|A(z)| \leq A(1)(1 - d'\epsilon^2) \quad \text{avec} \quad d' = \delta^2 d$$

Il ne nous reste plus qu'à montrer que dans ce cas le calcul de I_1 n'est pas modifié.

Calculons les contributions sur un $\mathcal{C}_{\delta,j}$, on a :

$$[z^n]\{A(z)^{\lambda n}\} = \int_{\mathcal{C}_{\delta,j}} A(z)^{\lambda n} \frac{dz}{z^{n+1}}$$

Soit $z = \omega e^{i\theta}$ avec $\omega^\delta = 1$; on a alors

$$[z^n]\{A(z)^{\lambda n}\} = \frac{1}{2\pi\omega^n} \int_{-\epsilon}^{\epsilon} A(e^{i\theta})^{\lambda n} e^{-in\theta} d\theta \quad \text{car} \quad A(\omega e^{i\theta}) = A(e^{i\theta})$$

Or le coefficient considéré n est nécessairement un multiple de δ et donc nécessairement $\frac{1}{\omega^n} = 1$: on est donc ramené au calcul de I_1 dans la cas précédent, le résultat final étant multiplié par δ et, au niveau de la moyenne, par simplification, $\overline{b_{i,n,k}}$ ne change pas.

Passons à présent au dernier cas

3.3.1.3 $a_1 = 0$ et $\delta = 1$

$A(z)$ est du type

$$A(z) = a_0 + a_k z^k + \dots + a_l z^l + \dots \quad \text{avec} \quad a_0, a_k, a_l \neq 0 \quad \text{et} \quad \text{pgcd}(k, l) = 1$$

On a $\mathcal{C}_2 = \mathcal{C} / \mathcal{C}_{1,1}$ avec les notations précédentes. On va retrouver nos majorations sur \mathcal{C}_2 . En effet - pour $\theta \in \mathcal{C} / \bigcup_{j=1}^k \mathcal{C}_{k,j}$, on a

$$|A(z)| \leq |a_0 + a_k e^{ik\theta}| + A(1) - a_0 - a_k$$

et

$$|a_0 + a_k e^{ik\theta}| \leq |a_0 + a_k e^{ik\epsilon}| = a_0 + a_k - K\epsilon^2$$

car $\cos(k\theta) \leq \cos(k\epsilon)$, d'où encore $\exists d > 0$ tel que :

$$|A(z)| \leq A(1)(1 - d\epsilon^2)$$

et cette contribution est exponentiellement négligeable.

- lorsque $\theta \in \mathcal{C}_{k,j}$, intervalle critique, on sait comme $\text{pgcd}(k,l) = 1$ que $\mathcal{C}_{k,j} \cap \mathcal{C}_{l,j'} = \emptyset$ (sauf pour $j = k$ et $j' = l$ qui de toute façon est dans I_1), et donc

$$|A(z)| \leq |a_0 + a_l e^{i\theta}| + A(1) - a_0 - a_l$$

$$\Rightarrow |A(z)| \leq A(1)(1 - d\epsilon^2)$$

Donc dans ce cas-ci, dans tout \mathcal{C}_2 , I_2 est exponentiellement négligeable devant $A(1)^{\lambda n}$.

Nous avons donc bien montré dans tout les cas que I_2 est exponentiellement négligeable devant $A(1)^{\lambda n}$ (le cas où $\delta = \text{pgcd}(\text{exposants de } A(z)) > 1$ introduisant une constante multiplicative dans l'évaluation de $[z^n]\{A(z)^{\lambda n}\}$ qui se simplifie dans le calcul de $\overline{b_{i,n,k}}$). Nous pouvons donc poursuivre par

3.3.2 Calcul de I_1

On doit estimer

$$I_1 = \frac{1}{2\pi} \int_{-\epsilon}^{\epsilon} A(e^{i\theta})^{\lambda n} e^{-in\theta} d\theta$$

ϵ étant voisin de zéro, on développe

$$A(e^{i\theta}) = A(1) \exp\left(\frac{i\theta}{\lambda} - B_1 \theta^2 + iB_2 \theta^3 + O(\theta^4)\right)$$

les constantes réelles B_i se déterminant aisément en fonction de λ , $A(1)$, $A'(1)$, $A''(1)$... Notons la valeur de

$$B_1 = \frac{1}{2} \left(\frac{1}{\lambda} - \frac{1}{\lambda^2} + \frac{A''(1)}{A'(1)} \right)$$

et remarquons que $B_1 > 0$ puisque

$$2A(1)^2 B_1 = A''(1)A(1) - A'(1)^2 + A'(1)A(1) \sim (M - M')^2 c_{M'} c_M c_0^{M'+M} > 0$$

quand c_0 est grand; M' désignant l'arité la plus grande après M (on est dans le cas $m \neq M$). B_1 est d'ailleurs à un facteur multiplicatif près l'expression de la variance de la taille des termes.

D'où

$$I_1 = \frac{A(1)^{\lambda n}}{2\pi} \int_{-\epsilon}^{\epsilon} e^{-\lambda n B_1 \theta^2} e^{iC_3 n \theta^3 + O(n\theta^4)} d\theta$$

Remarquons que le choix de ϵ assure que $n\theta^3 = \frac{\ln^6 n}{\sqrt{n}}$ est bien petit. Or

$$\int_{\epsilon}^{\infty} e^{-bn\theta^2} \theta^m d\theta \sim n^{-\frac{m+1}{2}} n^{-\ln n}$$

est exponentiellement négligeable devant n on va donc pouvoir considérer les bornes d'intégration comme infinies et utiliser le fait que

$$\int_0^{\infty} e^{-bn\theta^2} \theta^{\alpha} d\theta = \frac{\Gamma(\frac{\alpha+1}{2})}{2(nb)^{\frac{\alpha+1}{2}}} = O(n^{-\frac{\alpha+1}{2}})$$

avec Γ la fonction d'Euler; ce qui nous permet de borner les termes d'erreur. On va mener le calcul à la précision $O(n^{-\frac{3}{2}})$; on a donc

$$I_1 = \frac{A(1)^{\lambda n}}{2\pi} \int_{-\infty}^{\infty} e^{-\lambda n B_1 \theta^2} e^{i C_3 n \theta^3 + O(n \theta^4)} d\theta$$

et $e^{i C_3 n \theta^3 + O(n \theta^4)} = 1 + i n C_3 \theta^3 + O(n \theta^4)$; le terme en θ^3 disparaît par symétrie et le $O(n \theta^4)$ donne bien un $O(n^{-\frac{3}{2}})$ dans la formule finale. D'où

$$\begin{aligned} I_1 &= \frac{A(1)^{\lambda n}}{2\pi} \left(\int_{-\infty}^{\infty} e^{-\lambda n B_1 \theta^2} d\theta + O(n^{-\frac{3}{2}}) \right) \\ &= \frac{A(1)^{\lambda n}}{2\pi} \left(\sqrt{\pi n \lambda B_1}^{-\frac{1}{2}} + O(n^{-\frac{3}{2}}) \right) \end{aligned}$$

3.3.3 Valeur de $a_{n,k}$

I_2 étant exponentiellement négligeable devant $A(1)^{\lambda n}$, avec la valeur de I_1 trouvée ci-avant, on obtient :

THÉORÈME 6 : La valeur de $a_{n,k} = [z^n] A(z)^k$ est pour $k = \lambda n$:

$$I = a_{n,k} = \frac{A(1)^{\lambda n}}{2\sqrt{\pi}} \left(\frac{1}{2} \left(1 - \frac{1}{\lambda} \right) + \frac{\lambda A''(1)}{2A(1)} \right)^{-\frac{1}{2}} n^{-\frac{1}{2}} \left(1 + O\left(\frac{1}{n}\right) \right)$$

quantité à multiplier par δ lorsque $A(z) = N(z^\delta)$.

3.4. Résultats pour le modèle de liste

Le théorème précédent nous permet de déterminer la taille moyenne des nœuds-mémoire du graphe de RETE et d'en déduire avec ce modèle de liste de termes le coût moyen de l'algorithme.

3.4.1 Expression de $\overline{b_{i,n,k}}$ pour $m \neq M$

En utilisant l'expression de $a_{n,k}$ ci-dessus (et en reprenant les calculs précédents avec $n' = n - \omega_i + m$ et $A(z)^{\lambda n' + \mu}$ avec μ constante), les résultats combinatoires des théorèmes 5 et 3 se simplifient pour donner :

THÉORÈME 7 : Le nombre moyen de l -uplets de faits filtrés en un nœud i du graphe de RETE est :

$$\overline{b_{i,n,k}} = c_0^{x_i} \frac{1}{A(1)^l} k^l \left(1 + O\left(\frac{1}{n}\right) \right) = c_0^{x_i} \frac{\lambda^l}{A(1)^l} n^l \left(1 + O\left(\frac{1}{n}\right) \right) = \Pi_i n^l \left(1 + O\left(\frac{1}{n}\right) \right)$$

avec $l = 1$ dans le cas d'un nœud de l'arbre de discrimination.

Ce qui nous permet d'associer à un nœud du graphe de RETE son *taux de filtrage moyen* qui traduit la probabilité pour un terme ou un l -uplet de termes d'être filtré en un nœud i du graphe de RETE, donné par

THÉOREME 8 : Le taux de filtrage moyen en un nœud i du graphe de RETE pour le modèle de liste de termes est :

$$\overline{\alpha_i} = \frac{\overline{b_{i,n,k}}}{k^l} \simeq \frac{c_0^{x_i}}{A(1)^l}$$

Remarque : Notons bien sûr que $\overline{\alpha_i} \leq 1$ et on vérifie bien, au premier niveau de l'arbre de discrimination (nœuds 1, 2 et 3 de la figure 2 par exemple), lorsque l'on somme sur tous les symboles de tête possibles de $TP(F)$, que

$$\sum_i \overline{\alpha_i} = \sum_i \frac{c_i c_0^i}{A(1)} = 1 !$$

Il nous reste encore un petit cas à traiter :

3.4.2 Evaluation de $\overline{b_{i,n,k}}$ quand $m = M$

Ce cas se traite directement sans analyse asymptotique. On a donc c_0 et c_M et $P(z) = c_M c_0^M z^{M+1}$ et $\frac{1}{\lambda_u} = M + 1$. Pour un nœud i de l'arbre de discrimination la série génératrice des termes de $TP(F)$ qui filtrent en i est $B_i(z) = c_0^{x_i} z^{M+1}$. D'où

$$\begin{aligned} \overline{b_{i,n,k}} &= \frac{[z^n] \{k A(z)^{k-1} c_0^{x_i} z^{M+1}\}}{[z^n] A(z)^k} \quad \text{avec } n = (M+1)k \\ &= k \frac{c_0^{x_i}}{c_M c_0^M} \end{aligned}$$

Et, en suivant le raisonnement du théorème 5, pour un nœud i du graphe de jointure, on obtient :

$$\overline{b_{i,n,k}} = k^l \frac{c_0^{x_i}}{c_M^l c_0^{lM}}$$

et l'on s'aperçoit, ce qui confirme notre analyse asymptotique, que ces deux formules sont des cas particuliers exacts des expressions des théorèmes 7 et 8 précédents. Ceux-ci fournissent donc dans tous les cas du modèle de liste, l'estimation de la taille moyenne des nœuds-mémoire du graphe de RETE. On peut donc à présent conclure avec ce modèle :

3.4.3 Coût moyen de l'algorithme dans le modèle de liste

Déterminons le coût moyen de l'algorithme RETE, étant donné une Base de Faits quelconque et un graphe de RETE fixé. Nous allons évaluer le temps moyen nécessaire à l'algorithme RETE pour déterminer l'ensemble de conflit à partir d'une Base de Faits d'entrée donnée. Rappelons que notre mesure de complexité est le temps de réalisation d'un test multi-motifs; on peut, avec l'estimation de la taille moyenne des nœuds-mémoire, déterminer le nombre moyen de ces tests multi-motifs réalisés aux nœuds-jointure (ce qui représente comme on l'a déjà dit la partie principale du temps total d'exécution). Notons l_i le nombre de composantes des uplets de sortie en un nœud-jointure i et β_i le nombre de tests multi-motifs réalisés en i en considérant seulement un l_i -uplet en entrée (dans notre exemple $\beta_{17} = 2$, $\beta_{14} = 0$, $\beta_{13} = 1 \dots$). Le nombre moyen de tests réalisés en i est le nombre moyen de l_i -uplets testés en i multipliés par β_i . Pour obtenir le nombre moyen de l_i -uplets qui arrive en entrée en i il suffit de considérer qu'en i aucun test multi-motifs n'est réalisé (c'est à dire $y_i = 0$) et d'évaluer grâce au théorème 7 le $\overline{b'_{i,n,k}}$ ainsi modifié. Notons $\Pi_i n^{l_i}$ cette quantité

(remarquons que celle-ci est précisément égale à $\overline{b_{i1,n,k}} \times \overline{b_{i2,n,k}}$ où $i1$ et $i2$ désignent les deux nœuds entrées gauche et droite de i).

On peut alors proposer

$$\overline{\text{coût}_n} \simeq K \sum_{i \text{ nd-jt}} \beta_i \Pi'_i n^{l_i} = K \sum_{i \text{ nd-jt}} \beta_i (\Pi_{i1} \Pi_{i2}) n^{l_i} \quad (4)$$

avec K une constante dépendant de l'implémentation.

Bien sûr, asymptotiquement en n , on obtient la partie prépondérante de cette expression en gardant seulement les nœuds qui ont le plus grand l_i . Mais l'intérêt d'une analyse de complexité en moyenne est la détermination des constantes de proportionnalité, qui peuvent être pour l_i grand, comme on le verra en Annexe, très petites et donc fondamentales.

Remarque : En hachant les mémoires locales suivant les tests d'égalité réalisés aux nœuds-jointure, le nombre moyen de semi-unifications réalisées en un nœud-jointure i est seulement le nombre de l_i -uplets de sortie, c'est à dire $\overline{b_{i,n,k}}$. Et donc dans ce cas nous avons juste à remplacer dans la formule ci-dessus Π'_i par Π_i .

A partir de cette expression, qui peut être complètement précisée sur n'importe quel exemple, on peut émettre *a posteriori* les hypothèses de Forgy (cf. [Forg 79]) nécessaire à son estimation du coût dans le cas le pire de l'algorithme **RETE**. Par exemple, si l'on suppose que le nombre de motifs-condition par membre gauche dans la Base de Règles est constant égal à c , le résultat précédent devient :

$$\overline{\text{coût}_{|BF|}} \simeq K \lambda^c |BF|^c \left(\sum_{j=1}^{|BR|} \beta_j \Pi'_j \right) = K \lambda^c |BF|^c \left(\sum_{j=1}^{|BR|} \beta_j \Pi_{j1} \Pi_{j2} \right)$$

avec $j1$ et $j2$ les deux entrées du dernier nœud-jointure j de la règle Rj .

Et l'on retrouve le même type de relations proposées par Forgy : un coût linéaire en fonction du nombre de règles et un coût polynômial en fonction de la taille de la Base de Faits $|BF|^c$ (on a déterminé le temps d'exécution de l'algorithme **RETE** avec en entrée du graphe $|BF|$ termes-modification, Forgy trouvait un temps en $|BF|^{c-1}$ parce qu'il considérait seulement un terme-modification comme entrée).

IV. Calculs dans le cadre du modèle du multiensemble

Dans cette partie on va déterminer la *taille* moyenne des nœuds-mémoire du graphe de **RETE** en modélisant la Base de Faits par un multiensemble de faits. On présente tout d'abord (4.1) le modèle considéré et on détermine combinatoirement les quantités fondamentales; dont à nouveau on simplifie les expressions à l'aide d'une étude asymptotique (4.2); enfin on présente les résultats dans le cadre de ce modèle (4.3).

4.1. Combinatoire du multiensemble

Nous choisissons ici de modéliser la Base de Faits par un ensemble de termes non ordonné autorisant les répétitions (ceci pour traduire toujours les ajouts et les suppressions d'un même terme au cours des cycles du moteur d'inférence). La structure précédente de liste insistait sur le caractère successif des introductions de termes dans le graphe de **RETE**. En ce qui concerne le modèle du multiensemble, nous allons pouvoir effectuer des moyennes sur une Base de Faits quelconque indépendamment de son nombre d'éléments. Le modèle du multiensemble insiste sur l'indépendance logique de l'état

des mémoires locales du graphe de l'ordre d'ajout des faits de la Base de Faits. Nous comparerons bien sûr les résultats obtenus dans la section (4.3).

Il n'est plus nécessaire ici d'introduire une nouvelle notion de taille des termes et la série génératrice décrivant $TP(F)$ est donc :

$$P(z) = \sum_{t \in TP(F)} z^{|t|} = \sum_{n \geq 0} p_n z^n = \sum_{l=m+1}^{M+1} (c_{l-1} c_0^{l-1}) z^l$$

Nous considérons comme entrée du graphe de **RETE**, une Base de Faits de taille globale n . La série génératrice des multiensembles de termes de $TP(F)$ déterminée par $P(z)$ est donnée par :

$$M(z) = \prod_{t \in TP(F)} \left(\frac{1}{1 - z^{|t|}} \right) = \prod_{i=m}^{i=M} \left(\frac{1}{1 - z^{i+1}} \right)^{p_i}$$

Ainsi $[z^n]M(z)$, représente le nombre de multiensembles de taille globale n d'éléments de $TP(F)$.

Exemple : Dans notre exemple

$$M(z) = \left(\frac{1}{1 - z^3} \right)^{72} \left(\frac{1}{1 - z^4} \right)^{216}$$

Pour un nœud i du graphe de **RETE**, les notations $B_i(z)$, ω_i , x_i et y_i sont conservées. On a donc l'analogie aux théorèmes 1 et 2 :

THÉORÈME 9 : La série génératrice des l -uplets de termes qui sont filtrés en un nœud i du graphe de **RETE** est :

$$B_i(z) = c_0^{x_i} z^{\omega_i + 1}$$

avec $l = 1$ dans le cas d'un nœud de l'arbre de discrimination.

Déduisons de ceci le nombre moyen $\overline{b_{i,n}}$ de termes filtrés en i , nœud de l'arbre de discrimination, en considérant comme donnée de l'algorithme **RETE** une Base de Faits quelconque de taille n .

THÉORÈME 10 : Le nombre moyen de termes filtrés en un nœud i de l'arbre de discrimination est :

$$\overline{b_{i,n}} = \frac{c_0^{x_i} [z^{n-\omega_i-1}] \left\{ \frac{M(z)}{1 - z^{\omega_i+1}} \right\}}{[z^n]M(z)}$$

PREUVE : Nous allons faire une démonstration analogue à celle du théorème 3. On sépare $P(z)$ en : $P(z) = B_i(z) + R_i(z)$ avec $R_i(z)$ la "série reste" c'est à dire la série génératrice de tous les termes qui ne s'unifient pas avec le i -motif. Alors, on marque avec la variable u les éléments filtrés en i et on pose :

$$\begin{aligned} P(z, u) &= u B_i(z) + R_i(z) = P(z) + (u - 1) B_i(z) \\ &= \sum_{j=m, j \neq \omega_i+1}^M p_j z^j - c_0^{x_i} z^{\omega_i+1} + u c_0^{x_i} z^{\omega_i+1} \end{aligned}$$

Et en posant

$$M(z, u) = M(z) \frac{1}{(1 - z\omega_i + 1)^{-c_0^{x_i}}} \frac{1}{(1 - uz\omega_i + 1)^{c_0^{x_i}}}$$

$[u^p z^n]M(z, u)$ représente le nombre de multiensembles de taille n parmi lesquels il y a exactement p termes qui filtrent en i de l'arbre de discrimination.

On différencie alors de façon identique à la démonstration du théorème 3 et on obtient le résultat annoncé. ■

Pour un nœud-jointure i associés à l nœuds j^i de l'arbre de discrimination, l'ordre des tailles des mémoires locales en sortie des j^i n'est pas indifférent car il correspond à l'instanciation de motifs différents. Donc sémantiquement, l'ordre intervient pour la combinatoire des nœuds-jointure. Le raisonnement amenant au théorème 5 est donc valable et le nombre moyen de l -uplets de termes filtrant en un nœud i du graphe de jointure est encore :

$$\overline{b_{i,n}} = c_0^{y_i} \left(\prod_{j^i=1}^l \overline{b_{j^i,n}} \right)$$

Afin de simplifier les expressions obtenues, nous devons donc analyser asymptotiquement des fractions rationnelles particulières.

4.2. Analyse asymptotique

Nous devons donc étudier le comportement de $m_n = [z^n] \prod_j \frac{1}{(1-z)^{p_j}}$. Les singularités d'une telle fraction rationnelle sont toutes de module 1 et les contributions des différentes singularités s'ajoutent: nous allons distinguer deux cas suivant la valeur de $\delta = \text{pgcd}(p_j)$ c'est à dire le *pgcd* de l'arité des symboles de tête + 1.

4.2.1 $\delta = 1$

Dans ce cas 1 est la singularité de plus grande multiplicité $\sum_j p_j$. Et donc comme

$$[z^n] \frac{1}{(1-z)^\alpha} = \frac{n^{\alpha-1}}{\Gamma(\alpha)} \left(1 + O\left(\frac{1}{n}\right) \right)$$

1 va donner la contribution dominante dans m_n (la constante multiplicative étant donnée par le reste évalué en 1) et donc

$$\begin{aligned} m_n &= \frac{M(z)}{(1-z)^{\sum_j p_j}} \Big|_{z=1} \frac{n^{\sum_j p_j - 1}}{\Gamma(\sum_j p_j)} \left(1 + O\left(\frac{1}{n}\right) \right) \\ &= \prod_j \frac{1}{j^{p_j}} \frac{n^{\sum_j p_j - 1}}{\Gamma(\sum_j p_j)} \left(1 + O\left(\frac{1}{n}\right) \right) \end{aligned}$$

4.2.2 $\delta > 1$

Dans ce cas, on cherche $m_n = [z^n] \prod_l \frac{1}{(1-z^l)^{p_{\delta l}}}$. Les racines $\delta^{i\text{ème}}$ de l'unité sont ici de même importance et on a $m_n = 0$ si δ ne divise pas n et si $n = k\delta$:

$$\begin{aligned} m_n &= [z^k] \prod_l \frac{1}{(1-z^l)^{p_{\delta l}}} = \prod_l \frac{1}{l^{p_{\delta l}}} \frac{k^{\sum_l p_{\delta l} - 1}}{\Gamma(\sum_l p_{\delta l})} \\ &= \delta \prod_j \frac{1}{j^{p_j}} \frac{n^{\sum_j p_j - 1}}{\Gamma(\sum_j p_j)} \end{aligned}$$

Et comme auparavant, ce cas-ci ne modifie pas les résultats moyen à venir car le *pgcd* δ se simplifie dans l'expression de $\overline{b_{i,n}}$ (sauf dans le cas rarissime où $c_0 = 1$ (une seule constante !) et où le symbole de tête du i -motif est le seul de son arité et où la division de $M(z)$ par $(1 - z^{\omega_i+1})$ augmente la valeur de δ : cas fortement improbable).

4.3. Résultats avec le modèle du multiensemble

Grâce aux formules précédentes, on obtient donc en modélisant la Base de Faits par un multi-ensemble de termes :

THÉORÈME 11 : Pour une Base de Faits quelconque en entrée de taille n , en un nœud i
- de l'arbre de discrimination, le nombre moyen de termes filtrés vaut

$$\overline{b_{i,n}} = c_0^{x_i} \frac{1}{P(1)} \frac{1}{\omega_i + 1} n \left(1 + O\left(\frac{1}{n}\right) \right) = \Phi_i n \left(1 + O\left(\frac{1}{n}\right) \right)$$

- du graphe de jointure, le nombre moyen de l -uplets de termes filtrés vaut

$$\overline{b_{i,n}} = c_0^{x_i} \frac{1}{P(1)^l} \frac{1}{\prod_{j=1}^l (\omega_{j_i} + 1)} n^l \left(1 + O\left(\frac{1}{n}\right) \right) = \Phi_i n^l \left(1 + O\left(\frac{1}{n}\right) \right)$$

avec j^i les l nœuds de l'arbre de discrimination associés à i .

Comme on peut toujours relier le nombre d'éléments d'une Base de Faits à sa taille globale par $k \simeq \lambda_u n$, on peut définir une probabilité de présence moyenne :

THÉORÈME 12 : Le taux de filtrage d'un nœud i de l'arbre de discrimination est

$$\overline{\mu_i} \simeq c_0^{x_i} \frac{1}{P(1)} \frac{1}{\omega_i + 1} \frac{1}{\lambda_u}$$

et le taux de filtrage d'un nœud i du graphe de jointure est

$$\overline{\mu_i} \simeq c_0^{x_i} \frac{1}{P(1)^l} \frac{1}{\prod_{j=1}^l (\omega_{j_i} + 1)} \frac{1}{\lambda_u^l}$$

On voit donc que le modèle du multiensemble conduit au même type d'expression pour le coût global de l'algorithme RETE en remplaçant Π_i par Φ_i .

Les résultats obtenus dans le cadre du modèle de liste et du multiensemble de termes se comparent aisément (rappelons que $A(1) = P(1)$). Dans le second cas il apparaît un facteur correctif de la taille des termes filtrés par rapport à la taille moyenne; si l'on note $\sigma = \frac{1}{\lambda_u}$ la taille moyenne des termes alors

$$\overline{\mu_i} = \overline{\alpha_i} \prod_{j=1}^l \left(\frac{\sigma}{\omega_{j_i} + 1} \right)$$

$\omega_{j_i} + 1$ étant la taille des termes filtrés au nœud j^i .

V. Domaines de variation multiples et probabilités

Jusqu'à présent nous avons considéré que les constantes de $TP(F)$ prenaient indifféremment leur valeur dans un ensemble de cardinal c_0 ; ainsi nous considérons que celles-ci pouvaient instancier n'importe quel argument de n'importe quel symbole de tête. Or on sait que pour les Bases de Faits réelles les symboles de tête et les constantes ont une valeur sémantique qui interdit n'importe quelle instanciation. On peut avoir "(Mange singe bananes)" mais pas (sauf dans le cas de fruits carnivores !) "(Mange bananes singe)".

D'autre part, on sait aussi que dans les Bases de Faits réelles les symboles (*i.e.* les symboles de tête et les constantes) ont rarement des probabilités de présence uniforme comme nous l'avons supposé jusqu'à présent.

Dans cette partie nous allons donc généraliser les études précédentes de façon à prendre en compte ces deux faits. Nous allons introduire des *domaines de variations différents pour les constantes* (5.1) et des *probabilités pour les symboles de $TP(F)$* (5.2).

5.1. Domaines de variation multiples

Nous allons donc pour chaque symbole de tête différencier les domaines de variation de ses arguments. On se donne p domaines possibles de variation pour les constantes D_1, \dots, D_p de cardinalité respective d_1, \dots, d_p . On va supposer ces domaines disjoints ou plus précisément qu'il ne peut y avoir de tests d'égalité entre arguments que si ces deux arguments appartiennent au même domaine de variation. On va préciser chaque symbole de tête.

Soit un nœud i de l'arbre de discrimination. Le symbole de tête f_i du i -motif a pour arité ω_i et nous notons $f_{i,r}$ le nombre d'arguments variables de f_i qui varient dans le $r^{ième}$ domaine. On se donne donc pour chaque symbole de tête un p -uplet $f_{i,1}, \dots, f_{i,p}$ qui caractérise les variations des ses arguments (on a $\sum_{r=1}^p f_{i,r} = \omega_i$ et bien sûr certains $f_{i,r}$ peuvent être nuls).

Cette précision va entraîner des modifications mineures des résultats précédents. Ainsi

$$P(z) = \sum_{f_i \in F} \left(\prod_{r=1}^p d_r^{f_{i,r}} \right) z^{\omega_i+1} = z^{m+1} A(z)$$

Remarque : s'il n'y a qu'un seul domaine de cardinal c_0 , on retrouve $[z^n]P(z) = c_{n-1}c_0^{n-1}$.

On a toujours $\frac{1}{\lambda_u} = \frac{P'(1)}{P(1)}$ mais x_i est remplacé par un p -uplet $x_{i,1}, \dots, x_{i,p}$ avec $x_{i,r}$ qui désigne le nombre d'arguments *distincts* du i -motif appartenant à D_r . D'où

THÉORÈME 13 : La série génératrice des termes filtrés en un nœud i de l'arbre de discrimination est :

$$B_i(z) = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) z^{\omega_i-m}$$

ceci pour le modèle de liste, seule la puissance de z changeant pour le modèle du multiensemble.

Exemple : Considérons le motif :

$$(f \ a \ b \ a \ X \ Y \ X' \ X \ Z \ X)$$

avec $X, X' \in D_1, Y \in D_2$ et $Z \in D_3$ et a, b des constantes. On a alors $x_{i,1} = 2, x_{i,2} = 1$ et $x_{i,3} = 1$ et donc $B_i(z) = (d_1^2 d_2 d_3) z^{9-m}$.

On a bien sûr le nombre de variables distinctes du i -motif $= x_i$ précédent $= \sum_{r=1}^p x_{i,r}$.

Dans les formules combinatoires en un nœud i de l'arbre de discrimination, seule change la constante multiplicative $c_0^{x_i}$ qui est remplacée par $(\prod_{r=1}^p d_r^{x_{i,r}})$. Pour un nœud i du graphe de jointure, on va noter $x_{i,r} = \sum_{j=1}^l x_{j^i,r} + y_{i,r}$ avec $y_{i,r} (\leq 0)$ le terme correctif dû aux jointures dans D_r au nœud i . Avec cette convention, on voit facilement que rien n'est modifié des calculs précédents et seule la constante multiplicative $c_0^{x_i}$ est remplacée encore par $(\prod_{r=1}^p d_r^{x_{i,r}})$.

On a donc les résultats suivants en considérant des domaines de variation différents pour les arguments des symboles de tête :

THÉORÈME 14 : *Pour le modèle de liste, le nombre moyen de l -uplets de faits filtrés en un nœud i du graphe de RETE est :*

$$\begin{aligned} \overline{b_{i,n,k}} &= \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{1}{A(1)^l} k^l \left(1 + O\left(\frac{1}{n}\right) \right) = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{\lambda^l}{A(1)^l} n^l \left(1 + O\left(\frac{1}{n}\right) \right) \\ &= \Theta_i n^l \left(1 + O\left(\frac{1}{n}\right) \right) \end{aligned}$$

et le taux de filtrage moyen en un nœud i du graphe de RETE est :

$$\overline{\tau_i} \simeq \frac{(\prod_{r=1}^p d_r^{x_{i,r}})}{A(1)^l}$$

Pour le modèle de multiensemble, pour un nœud i

- de l'arbre de discrimination, le nombre moyen de termes filtrés en i vaut

$$\overline{b_{i,n}} = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{1}{P(1)^l} \frac{1}{\omega_i + 1} n \left(1 + O\left(\frac{1}{n}\right) \right) = \Psi_i n \left(1 + O\left(\frac{1}{n}\right) \right)$$

- du graphe de jointure, le nombre moyen de l -uplets de termes filtrés en i vaut

$$\overline{b_{i,n}} = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{1}{P(1)^l} \frac{1}{\prod_{j=1}^l (\omega_{j^i} + 1)} n^l \left(1 + O\left(\frac{1}{n}\right) \right) = \Psi_i n^l \left(1 + O\left(\frac{1}{n}\right) \right)$$

avec j^i les l nœuds de l'arbre de discrimination associés à i .

Le taux de filtrage en un nœud i de l'arbre de discrimination est

$$\overline{\xi_i} \simeq \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{1}{P(1)^l} \frac{1}{\omega_i + 1} \frac{1}{\lambda_u}$$

et en un nœud i du graphe de jointure

$$\overline{\xi_i} \simeq \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{1}{P(1)^l} \frac{1}{\prod_{j=1}^l (\omega_{j^i} + 1)} \frac{1}{\lambda_u^l}$$

Là encore, on voit que le coût global de l'algorithme **RETE** n'est modifié qu'en remplaçant Π_i par Ψ_i ou par Θ_i .

Remarquons aussi que le cas où il faut se soucier de l'apparition d'un *pgcd* dans les résultats du modèle du multiensemble est encore plus rare puisque la condition $c_0 = 1$ est ici remplacée par $\forall r, d_r = 1$.

Cette notion de domaine de variation pour les constantes va également nous permettre d'étendre légèrement la classe des tests étudiés aux nœuds du graphe de **RETE**. Jusqu'à présent nous ne considérons que des tests d'égalité entre variables ou entre variable et constante. Nous avons déjà dit que la négation serait étudiée dans la partie 6. Mais ici on peut envisager de traiter un type de *test d'inégalité*. Si en un nœud i , on doit tester $X > a$; ceci suppose que le domaine D auquel appartiennent X et a est ordonné. On est donc en mesure de déterminer le cardinal d_a de $D_a = \{ Y \in D \mid Y > a \}$. Ce test d'inégalité va donc filtrer d_a fois plus d'éléments qu'un test du type $X = b$ avec $b \in D$. Son coût par rapport à ce dernier test d'égalité classique n'a donc qu'à être multiplié par d_a .

5.2. Modèle probabiliste

Jusqu'à présent on considérait une distribution uniforme sur tous les termes de même taille n . Afin d'obtenir une modélisation plus fine et plus vraisemblable, nous allons considérer un modèle prenant en compte le fait que pour beaucoup d'applications certains symboles sont plus fréquents que d'autres (Cf. [Alb 88], [AF 88] et [FSS]). On va considérer une *fonction de poids* w qui associe à chaque symbole de tête $f \in F$ (resp. à chaque constante c dans chaque domaine) un réel positif $w[f]$ (resp. $w[c]$). Ce poids s'étend multiplicativement aux termes. Si t est un terme, son poids $w[t]$ est défini comme

$$w[t] = \prod_{\text{symbole} \in t} w[\text{symbole}]$$

Pour utiliser un modèle de probabilité pour les symboles, on prend

$$\sum_{f \in F} w[f] = |F| \quad \text{et} \quad \sum_{c \in D_r} w[c] = d_r \quad \forall r$$

($|F|$ étant le cardinal de F). On a préféré considérer des *pondérations* et non des probabilités (dont la somme aurait fait 1) afin de rester compatible avec les domaines de variables multiples et conserver des résultats semblables.

THÉORÈME 15 : La série génératrice qui va décrire notre ensemble de termes ainsi pondérés est donc

$$W(z) = \sum_{t \in TP(F,w)} w(t) z^{|t|} = \sum_{f_i \in F} w(f_i) \left(\prod_{r=1}^p d_r^{f_{i,r}} \right) z^{\omega_i+1}$$

(pour la taille usuelle par exemple).

PREUVE : En effet lorsque l'on somme $w(t)$ pour tous les termes t ayant f_i comme symbole de tête, $w(f_i)$ se met bien en facteur et $\left(\prod_{r=1}^p d_r^{f_{i,r}} \right)$ réapparaît :

$$\begin{aligned} \sum_{t \text{ avec } f_i} w(t) &= w(f_i) \left(\sum_{c_{i,r} \in D_r} \left(\prod_{r=1}^p w(c_{i,r})^{f_{i,r}} \right) \right) = w(f_i) \prod_{r=1}^p \left(\sum_{c_{i,r} \in D_r} w(c_{i,r})^{f_{i,r}} \right) \\ &= w(f_i) \left(\prod_{r=1}^p d_r^{f_{i,r}} \right) \end{aligned}$$

La seule modification par rapport aux résultats précédents va donc être l'apparition d'une constante multiplicative correspondant au i -motif. Ainsi

THÉORÈME 16 : *La série génératrice des termes (pour la taille usuelle) filtrés en nœud i de l'arbre de discrimination est, en conservant les notations précédentes :*

$$B_i(z) = \sum_{t \text{ filtré en } i} w(t)z^{|t|} = W_i \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) z^{\omega_i+1}$$

avec W_i le poids du i -motif.

Plus précisément W_i égal à $w(f_i)$ multiplié par le poids des arguments déjà instanciés (les variables ayant un poids 1 par convention). Ce poids s'étend naturellement multiplicativement pour les nœuds-jointure aux l -uplets de termes et $W_i = \prod_{j=1}^l W_{j^i}$.

Il apparaît donc clairement que la seule modification apportée aux résultats de la section précédente pour $\overline{b_{i,n,k}}$, $\overline{b_{i,n}}$ et par conséquent pour le taux de filtrage moyen et pour le coût global de l'algorithme va être la présence de cette constante multiplicative W_i traduisant le poids du i -motif. On a par exemple les taux de filtrage moyen suivant le modèle considéré

$$W_i \frac{(\prod_{r=1}^p d_r^{x_{i,r}})}{A(1)^l} \quad \text{ou} \quad W_i \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{1}{P(1)^l} \frac{1}{\prod_{j=1}^l (\omega_{j^i} + 1)} \frac{1}{\lambda_u^l}$$

Quant au choix de ces pondérations, celui-ci peut être influencé par une Base de Faits particulière à traiter en dénombrant donc *statiquement* des fréquences d'apparition; ou plus *dynamiquement* en considérant les membres droits des règles de réécriture qui vont modifier la Base de Faits initiale au cours des cycles du moteur d'inférence et en pondérant suivant l'ajout ou la suppression de tel fait ou motif lorsque le membre gauche est instancié.

VI. La négation

Nous pouvons aisément étendre les calculs précédents à un modèle prenant en compte la *négation*. Plus précisément nous pouvons considérer la négation de motifs-condition dans le membre gauche des règles ou aussi des tests de non-égalité entre variables. Nous avons expliqué en fin de section (2.1) l'implémentation du nœud-jointure NON; il faut noter que la longueur des uplets en sortie du nœud est celle des uplets en entrée gauche.

Les résultats que l'on a obtenus sont assez souples, par exemple si l'on veut calculer le nombre moyen de termes filtrés par le motif suivant (avec *Tête* un symbole de tête particulier) :

$$(Tête \ X \neg X)$$

celui-ci est égal au nombre moyen de termes filtrés par $(Tête \ X \ Y)$ moins le nombre moyen de termes filtrés par $(Tête \ X \ X)$ (ce qui s'étend combinatoirement facilement au cas de plusieurs variables niées simultanément).

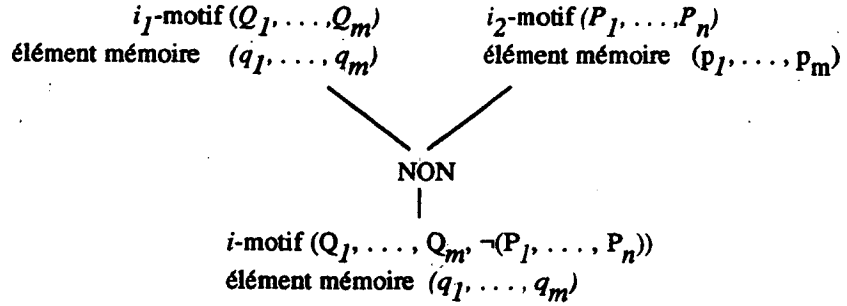


Figure 3 : Jointure NON

Considérons à présent la négation d'un motif-condition ou d'un l -uplet de motifs-condition. Pour un nœud i (cf. figure 3), on connaît le taux de filtrage moyen $\overline{\varphi}_i$ (à remplacer suivant le modèle et le degré de raffinement par $\overline{\alpha}_i, \overline{\mu}_i, \overline{\tau}_i, \overline{\xi}_i \dots$) aux nœuds i_1 et i_2 . Le taux de filtrage du motif $\neg(P_1, \dots, P_n)$ est $1 - \overline{\varphi}_{i_2}$ et donc le taux de filtrage du nœud i correspondant au i -motif $(Q_1, \dots, Q_m, \neg(P_1, \dots, P_n))$ est $\overline{\varphi}_i = \overline{\varphi}_{i_1}(1 - \overline{\varphi}_{i_2})$ (quand il n'y a pas de variables communes entre les P_i et les Q_j sinon on modifie dans cette formule $\overline{\varphi}_{i_2}$ avec le raisonnement ci-avant). Et les taux de filtrage des nœuds-jointure qui sont sous ce nœud (c'est à dire qui peuvent être atteints depuis i) sont modifiés de la même façon : on détermine normalement leur taux de filtrage en oubliant l'existence du uplet de motifs $\neg(P_1, \dots, P_n)$ et on multiplie le résultat obtenu par le coefficient $(1 - \overline{\varphi}_{i_2})$ pour obtenir le taux de filtrage réel.

Remarque : Le cas du nœud 12 de la figure 1 requiert les deux raisonnements ci-dessus à la fois.

Remarquons plus généralement que tous les résultats de cet article déterminent des quantités *localement* en un nœud i du graphe de RETE i.e. toutes les tailles locales, taux de filtrage... peuvent se calculer directement en ce nœud, connaissant ses paramètres caractéristiques, indépendamment du reste du graphe *cependant pratiquement* il apparaît plus commode de les calculer en propageant les résultats de façon descendante dans le graphe : ceci permet des calculs plus rapides (on calcule les produits cartésiens multipliés par les termes correctifs) et ceci implémente efficacement la négation sur une branche du graphe (la multiplication par $(1 - \overline{\varphi}_{i_2})$ se propage automatiquement).

Avant de conclure, nous allons montrer comment toute cette étude peut facilement s'appliquer à un domaine "voisin" du filtrage dans les systèmes experts :

VII. Requêtes multiples dans les Bases de Données Relationnelles

Le problème de l'estimation de la taille moyenne d'une équijointure de relations dans une Base de Données Relationnelle est un problème "voisin" de celui que nous venons de traiter. En effet, nous allons montrer dans cette partie, qu'avec un raisonnement analogue aux précédents, on peut déterminer la taille moyenne d'une équijointure (composée et après d'éventuelles sélections) étant donnée une Base de Données Relationnelle quelconque. Dans un premier temps nous introduirons le problème, adapterons les modèles précédents et préciserons quelques points de vocabulaire des Bases de Données Relationnelles (7.1). Puis nous développerons notre étude en effectuant une moyenne pour une Base de Données Relationnelle quelconque de k termes (7.2) et pour une Base de Données Relationnelle dont on sait estimer les tailles des relations initiales (7.3).

7.1. Modèle d'étude de l'équijointure

Transposons tout d'abord le vocabulaire de cet article en celui consacré des Bases de Données Relationnelles. Ainsi nous ne parlerons plus de symboles de tête mais de symboles de *relation*, notés $R[X, Y]$ (pour $(R \ X \ Y)$ auparavant). X et Y désignent toujours les arguments ou *attributs* de la relation. Un *tuple* instancie une relation comme un terme instancie un motif. Effectuer une *requête* dans une Base de Données Relationnelle signifie rechercher tous les tuples qui instancient une relation donnée. Les arguments d'une relation peuvent avoir une valeur constante : c'est l'opération de *sélection*. Les opérations de sélection correspondent à la discrimination effectuée dans l'arbre-test. Ainsi tous les tuples qui auront comme symbole de relation A et dont le premier attribut vaudra a_0 instancieront le motif $A[a_0, Y]$. Il faut également préciser ce que l'on appelle *taille d'une relation* : il s'agit du nombre de tuples qui l'instancient dans la Base de Données Relationnelle. Ce n'est donc pas la notion de taille des parties précédentes; il faut d'ailleurs souligner que souvent dans $R[X, Y]$, X désigne l'attribut de jointure et Y la *liste* de tous les attributs qui n'en font pas partie (ce n'est donc pas forcément une relation de taille usuelle 3).

Précisons également que le type de jointure que nous avons étudié jusqu'alors se nomme en Base de Données Relationnelle une *équijointure*; nous conserverons pour simplifier le terme de *jointure*. On note la jointure d'une relation A et d'une relation B sur l'attribut X par $A[X, Y] \bowtie B[X, Z]$ (ce qui correspond aux deux motifs en membre gauche $(A \ X \ Y) (B \ X \ Z)$). Nous allons déterminer la *taille moyenne d'une jointure* c'est à dire le nombre moyen de tuples (plus précisément ici de couples de tuples) qui instancient la relation dérivée. On pourra se reporter à [Rosen 81], [GP 84] et [GP 88]. En fait nous allons nous intéresser ici à l'estimation de la taille moyenne d'une *requête multiple* i.e. d'une *multi-jointure* après d'éventuelles *sélections*, par exemple chercher la taille moyenne de

$$R[a_0, X, Y] \bowtie S[a_1, X, Z] \bowtie T[Y, Z, U, b, Z'] \bowtie \dots$$

ce qui est l'analogue de l'instanciation d'un membre gauche d'une règle de réécriture.

7.2. Etude combinatoire pour une Base de Données de k tuples

Pour cela nous allons commencer par effectuer une moyenne pour un Base de Données Relationnelle quelconque de k tuples. Introduisons notre modèle combinatoire : nous disposons toujours d'un ensemble F de symboles de relation d'arité fixée et d'un ensemble de constantes C , les tuples étant construit comme pour $TP(F)$ et formant un ensemble noté $T(F)$. Considérons directement que les constantes sont prises dans des domaines de variation multiples D_1, \dots, D_p . On dispose, comme avant, pour un symbole de relation f_i du p -uplet des $f_{i,1}, \dots, f_{i,p}$ qui décrit les domaines de variation de ses attributs. Comme nous l'avons vu, nous ne disposons pas toujours de la taille "usuelle" d'un tuple et d'ailleurs celle-ci ne nous intéresse pas puisque nous cherchons seulement à déterminer le *nombre* de tuples qui instancient une relation; nous pouvons donc proposer une taille arbitraire de 1 à tous les tuples et comme série génératrice caractéristique

$$Q(z) = \sum_{f \in F} \left(\prod_{r=1}^p d_r^{f_{i,r}} \right) z$$

De même nous devons adapter notre modélisation de la Base de Données Relationnelle. En effet on évite les redondances dans les Bases de Données Relationnelles, il n'y a pas répétition d'un même tuple et on va modéliser la Base de Données Relationnelle par un *ensemble de tuples* et non par un multiensemble ou une liste. La série génératrice qui va décrire les ensembles de tuples éléments de $T(F)$ est donc :

$$E(z) = \prod_{t \in T(F)} (1 + z) = (1 + z)^{\sum_{f \in F} \left(\prod_{r=1}^p d_r^{f_{i,r}} \right)} = (1 + z)^{Q(1)}$$

On va à présent déterminer le nombre moyen de tuples quiinstancient une relation i . Pour cela il faut à nouveau définir les $x_{i,r}$ c'est à dire le p -uplet des nombres de variables distinctes de la relation dans les domaines D_r . On a alors :

THÉORÈME 17 : *La série génératrice des tuples instanciant la relation i est :*

$$B_i(z) = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) z$$

On peut alors donner

THÉORÈME 18 : *Le nombre moyen de tuples qui instancient une relation i , étant donnée une Base de Données Relationnelle quelconque de k tuples, est :*

$$\overline{b_{i,k}} = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{k}{Q(1)}$$

PREUVE : En effet, la preuve va se dérouler classiquement. On va distinguer par la variable u les tuples filtrés par la relation, on a

$$Q(z, u) = Q(z) + (u - 1)B_i(z) \quad \text{et} \quad B_i(1) = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right)$$

et

$$E(z, u) = \frac{E(z)}{(1+z)^{B_i(1)}} (1+uz)^{B_i(1)}$$

avec $[z^k u^q]E(z, u)$ le nombre de Bases de Données Relationnelles à k tuples parmi lesquels q exactement instancient i . On a donc

$$\overline{b_{i,k}} = \frac{[z^k] \frac{\partial}{\partial u} (E(z, u)) \big|_{u=1}}{[z^k] E(z)}$$

Or

$$\frac{\partial}{\partial u} (E(z, u)) \big|_{u=1} = z \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) E(z) (1+z)^{-1}$$

d'où

$$\overline{b_{i,k}} = B_i(1) \frac{[z^{k-1}] E(z) (1+z)^{-1}}{[z^k] E(z)} = B_i(1) \frac{\binom{Q(1)-1}{k-1}}{\binom{Q(1)}{k}} = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{k}{Q(1)}$$

Grâce à cette taille unique unité, nous n'avons pas eu besoin d'analyse asymptotique. ■

Lorsque l'on va considérer une multi-équijointure i , il va se produire une situation analogue à celle d'un nœud-jointure i : il faut introduire des facteurs correctifs $y_{i,r}$ dus à la jointure et le raisonnement du théorème 5 s'applique à nouveau.

THÉORÈME 19 : *Etant donnée une Base de Données Relationnelle de k tuples, le nombre moyen de l -uplets de tuples qui instancient une relation i dérivée d'une multi-équijointure (de l relations initiales) est :*

$$\overline{b_{i,n}} = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{k^l}{Q(1)^l}$$

et le taux d'instanciation d'une telle jointure est :

$$\overline{\varrho_i} = \frac{(\prod_{r=1}^p d_r^{x_{i,r}})}{Q(1)^l}$$

avec $x_{i,r}$ le nombre de variables distinctes de la relation dérivée dans le domaine D_r .

Les formules auxquelles nous sommes arrivés sont donc exactes et elles correspondent aux estimations obtenues dans le cadre du modèle du multi-ensemble. C'est donc une confirmation de l'analyse asymptotique précédente et cela indiquerait que la part des répétitions dans ce modèle précédent est négligeable en moyenne.

Il arrive souvent que l'on dispose d'informations plus précises sur la Base de Données Relationnelle initiale que son nombre de tuples et ceci peut permettre d'améliorer l'estimation moyenne de la taille d'une jointure :

7.3. Base de Données dont les tailles des relations sont connues

Souvent en effet on a une idée de la taille des relations de la Base de Données Relationnelle soit empiriquement soit on dispose de lois de distribution... On ne va donc plus effectuer des moyennes sur des Bases de Données Relationnelles quelconques de k tuples mais sur des Bases de Données Relationnelles qui vérifient ces tailles connues (il y a bien sûr plusieurs Bases de Données Relationnelles qui ont ces caractéristiques).

Supposons que dans notre Base de Données k_i soit la taille supposée de la relation f_i . La structure de chacune des relations est toujours donnée par le p -uplet des $f_{i,1}, \dots, f_{i,p}$ qui décrit les domaines de variation des attributs du symbole de relation f_i . On connaît $|F|$ le nombre de relations. On peut proposer comme série génératrice caractéristique de $T(F)$

$$Q(u_1, \dots, u_{|F|}) = \sum_{j=1}^{|F|} \left(\prod_{r=1}^p d_r^{f_{j,r}} \right) u_j$$

avec u_j qui distingue la taille maximum de la $j^{ième}$ relation. On peut à présent considérer les ensembles formés à partir de $T(F)$:

$$E(u_1, \dots, u_{|F|}) = \prod_{j=1}^{|F|} (1 + u_j)^{\prod_{r=1}^p d_r^{f_{j,r}}}$$

Comme les variables u_j marquent chacune une relation $[u_1^{q_1} \dots u_{|F|}^{q_{|F|}}] E(u_1, \dots, u_{|F|})$ représente le nombre de Bases de Données Relationnelles avec exactement q_j tuples instanciant la $j^{ième}$ relation.

Soit une relation dont le symbole est f_i et qui est caractérisée par son p -uplet $x_{i,1}, \dots, x_{i,p}$, $x_{i,r}$ étant le nombre de variables distinctes de la relation dans le domaine D_r . On a ici

THÉOREME 20 : La série génératrice des tuples instanciant la relation i est :

$$B_i(u_1, \dots, u_{|F|}) = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) u_i$$

et on trouve

THÉOREME 21 : Le nombre moyen de tuples qui instancient une relation i est :

$$\overline{b_{i,k_1, \dots, k_{|F|}}} = \left(\prod_{r=1}^p d_r^{x_{i,r} - f_{i,r}} \right) k_i$$

pour une Base de Données Relationnelle quelconque avec q_j tuples instanciant la $j^{i\grave{e}me}$ relation.

PREUVE : Remarquons tout d'abord que $x_{i,r} - f_{i,r} \leq 0$. La preuve est analogue à celle du théorème 18. Pour simplifier l'écriture, on notera le $|F|$ -uplet $(u_1, \dots, u_{|F|}) : (u_j)$. On va donc distinguer les tuples filtrés par la relation i par la variable u , on a

$$Q((u_j), u) = Q((u_j)) + (u - 1)B_i((u_j))$$

(il faut faire la différence entre la variable u_i qui compte les tuples dont le symbole de relation est f_i et la variable u qui compte ceux qui instancient la relation i où certains arguments peuvent être égaux ou encore constants à la suite de sélections).

On forme alors

$$E((u_j), u) = \frac{E((u_j))}{(1 + u_i)(\prod_{r=1}^p d_r^{x_{i,r}})} (1 + u u_i)(\prod_{r=1}^p d_r^{x_{i,r}})$$

avec $[u^q u_1^{q_1} \dots u_{|F|}^{q_{|F|}}] E((u_j), u)$ le nombre de Bases de Données Relationnelles avec q_j tuples de symbole de relation f_j et q tuples qui instancient i . D'où

$$\begin{aligned} \overline{b_{i,k_1, \dots, k_{|F|}}} &= \frac{[u_1^{k_1} \dots u_{|F|}^{k_{|F|}}] \frac{\partial}{\partial u} (E((u_j), u)) \big|_{u=1}}{[u_1^{k_1} \dots u_{|F|}^{k_{|F|}}] E((u_j))} \\ &= \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \left(\prod_{r=1}^p d_r^{f_{i,r} - 1} \right)^{k_i - 1} \prod_{j=1, j \neq i}^{|F|} \left(\prod_{r=1}^p d_r^{f_{j,r}} \right)^{k_j} \prod_{j=1}^{|F|} \left(\prod_{r=1}^p d_r^{f_{j,r}} \right)^{-1} \\ &= \left(\prod_{r=1}^p d_r^{x_{i,r} - f_{i,r}} \right) k_i \end{aligned}$$

Exemple : Pour la relation $A[a_0, Y]$, si l'on considère que les Bases de Données Relationnelles contiennent en moyenne k_a tuples dont le symbole de relation est A , si le domaine de variation de la première variable de A a pour taille d_1 et celui de Y d_2 , on a

$$\overline{b_{k_a}} = \frac{k_a}{d_1}$$

puisque $f_1 = 1$, $f_2 = 1$, $x_1 = 0$ et $x_2 = 1$.

Remarque : On voit par cet exemple et bien sûr par le théorème 21 que le nombre moyen de tuples qui instantient la relation (avec sélection et égalité entre variables éventuelles) ne dépend que de la taille de la relation initiale. *On peut se passer des tailles des autres symboles de relation dans la Base de Données Relationnelle initiale.*

Pour une multi-équijointure la taille moyenne de la relation dérivée va, toujours avec le raisonnement du théorème 5, être le produit des tailles moyennes des relations avant la jointure multiplié par un facteur correctif. On a ainsi :

THÉORÈME 22 : *Soit une Base de Données Relationnelle quelconque avec q_j tuples qui instantient la $j^{ième}$ relation. On considère une multi-équijointure i de l relations j^i initiales. Le nombre moyen de l -uplets de tuples qui instantient la relation i est :*

$$\overline{b_{i,k_1,\dots,k_l}} = \left(\prod_{r=1}^p d_r^{x_{i,r}} \right) \frac{\prod_{j^i=1}^l k_{j^i}}{\prod_{j^i=1}^l \prod_{r=1}^p d_r^{f_{j^i,r}}}$$

et le taux d'instanciation d'une telle jointure est :

$$\overline{v_i} = \frac{(\prod_{r=1}^p d_r^{x_{i,r}})}{\prod_{j^i=1}^l (\prod_{r=1}^p d_r^{f_{j^i,r}})}$$

avec $x_{i,r}$ le nombre de variables distinctes de la relation dérivée dans le domaine D_r .

Exemple : Considérons l'exemple classique de la taille moyenne de la jointure i :

$$R[X, Y] \bowtie S[X, Z]$$

On suppose connues les tailles initiales des relations $R[X, Y]$ et $S[X, Z]$ soient r et s . On connaît les cardinaux des domaines de variation des trois variables d_X , d_Y et d_Z . On a $x_{i,X} = 1$, $x_{i,Y} = 1$, $x_{i,Z} = 1$, $f_{R,X} = 1$, $f_{R,Y} = 1$, $f_{R,Z} = 0$, $f_{S,X} = 1$, $f_{S,Y} = 0$ et $f_{S,Z} = 1$ d'où

$$\overline{b_{i,r,s}} = d_X d_Y d_Z \frac{rs}{d_X d_Y d_X d_Z} = \frac{rs}{d_X}$$

et on retrouve la relation bien connue ([Rosen 81]).

Remarque : On note à nouveau que pour déterminer la taille moyenne de la jointure de l relations, on a besoin de connaître uniquement les tailles avant jointure et sélection des relations "concernées" et non pas de savoir le nombre des tuples commençant par chaque symbole de relation de la Base de Données Relationnelle.

Nous allons enfin détailler l'exemple de la section 7.1

$$R[a_0, X, Y] \bowtie S[a_1, X, Z] \bowtie T[Y, Z, U, b, Z']$$

Pour celui-ci, on dispose des tailles des domaines de variation d_X , d_Y , d_Z (Z et Z' sont deux variables distinctes qui varient dans le même domaine D_Z), d_U , d_a et d_b (qui sont les tailles des domaines de variation des constantes a_0 et a_1 , b respectivement). On connaît les nombres de tuples dont le symbole de relation est R , S ou T soient r , s ou t respectivement dans la Base de Données Relationnelle initiale. On a $f_{R,X} = 1$, $f_{R,Y} = 1$, $f_{R,Z} = 0$, $f_{R,U} = 0$, $f_{R,a} = 1$, $f_{R,b} = 0$, $f_{S,X} = 1$, $f_{S,Y} = 0$, $f_{S,Z} = 1$, $f_{S,U} = 0$, $f_{S,a} = 1$, $f_{S,b} = 0$, $f_{T,X} = 0$, $f_{T,Y} = 1$, $f_{T,Z} = 2$, $f_{T,U} = 1$, $f_{T,a} = 0$, $f_{T,b} = 1$. Enfin pour la relation dérivée i on a $x_{i,X} = 1$, $x_{i,Y} = 1$, $x_{i,Z} = 2$, $x_{i,U} = 1$, $x_{i,a} = 0$, $x_{i,b} = 0$. On en déduit donc d'après le théorème 22 :

$$\overline{b_{i,r,s,t}} = d_X d_Y d_Z^2 d_U \frac{rst}{d_a d_X d_Y d_a d_X d_Z d_Y d_Z^2 d_U d_b} = \frac{rst}{d_X d_Y d_Z d_a^2 d_b}$$

VIII. Conclusion

Nous avons précisé dans la section 2.2 la notion de coût moyen pour l'algorithme RETE. Cette complexité en moyenne est donnée par la formule (1) du 3.4.3. Cette formule nous donne le coût moyen en temps de l'algorithme RETE pour déterminer l'ensemble des règles instanciées à partir de la Base de Faits initiale et d'une Base de Règles fixée. Ce résultat est donné en fonction de la taille moyenne des mémoires locales du graphe de RETE. L'expression de cette quantité est donnée, suivant les modèles, dans les théorèmes : 7 pour le modèle de liste, 11 pour le modèle du multienemble, 14 lorsque l'on considère des domaines de variation multiples pour les constantes, 16 et fin de 5.2 lorsque l'on met des probabilités sur les symboles. Toutes ces expressions se calculent simplement en fonction des paramètres de n'importe quelle Base de Faits et Base de Règles.

Ces résultats théoriques sont confrontés à l'expérimentation sur le générateur de systèmes experts Xrete [LCR 88] pour lequel est développé un analyseur automatique de performance en moyenne à partir des résultats de cet article (des expérimentations analogues ont déjà été menées dans le cas du langage LISP par Clark, cf. [Clark 79]). Ces résultats théoriques nous permettent d'ailleurs de proposer, comme optimisation de l'algorithme, de réordonner les nœuds du graphe de RETE afin de diminuer les tailles des mémoires locales de la racine aux nœuds terminaux.

On a montré dans la section 5.1 et la partie 6 que l'on pouvait inclure dans notre étude non seulement des tests d'égalité aux nœuds du graphe de RETE mais encore la négation et un test d'inégalité. L'extension à des tests plus généraux encore est envisagée.

Les résultats obtenus s'étendent sans peine lorsque l'on effectue un hachage des mémoires locales ; ces résultats peuvent s'adapter à beaucoup de variantes de l'algorithme RETE (ARBRE D'UNIFICATION ..., cf. [Alb 88]).

Ces mêmes raisonnements s'appliquent en Base de Données Relationnelle. En effet, dans la partie 7 théorème 19, on estime la taille moyenne d'une multi-équijointure (après d'éventuelles sélections); ceci pour une Base de Données Relationnelle quelconque de k tuples. Et l'on précise ces résultats lorsque l'on connaît les tailles initiales des relations à joindre (7.3 théorème 22). Ces résultats sont également utilisés pour l'optimisation des requêtes dans les Bases de Données Relationnelles (système COSMA [RS 89]).

Le travail développé dans cet article montre une nouvelle fois la puissance et la souplesse de la théorie des fonctions génératrices dans l'analyse précise (et en moyenne) des algorithmes.

IX. Bibliographie

[Alb 87] L. Albert, "Présentation et évaluation de la complexité de l'algorithme RETE de multi-pattern matching dans les systèmes de règles de production", rapport de DEA, Université de PARIS XI, ENS, rapport de recherche 87-8 LCR Thomson-CSF, 1987.

[Alb 88] L. Albert, "Présentation et évaluation de la complexité en moyenne d'algorithmes de filtrage dans les moteurs d'inférences (Rete et arbre d'unification)", *Revue d'intelligence artificielle*, volume 2, Hermes, 1988, pp. 7-40.

[AF 88] L. Albert, F. Fages, "Average case complexity analysis of the RETE pattern match algorithm", proceedings of the 15th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, 317, Springer-Verlag, Tampere, Finland, July 1988.

[dB 58] NG de Bruijn, *Asymptotic Methods in Analysis*, Dover 1958.

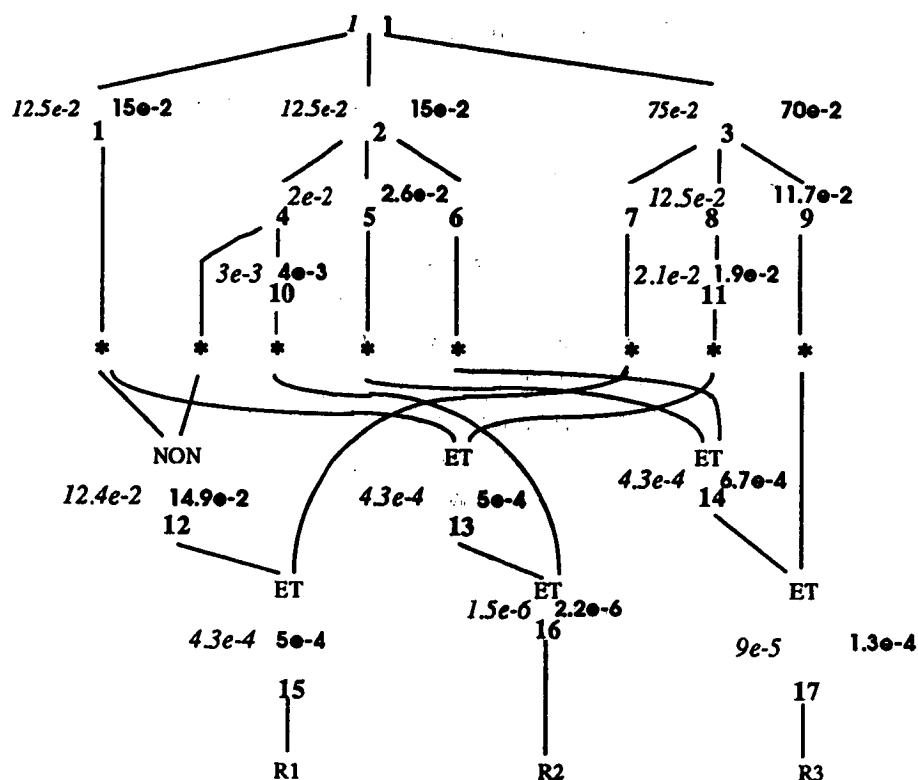
[CKS 86] C. Choppy, S. Kaplan, M. Soria, "Algorithmic complexity of term rewriting systems", Proceedings of the First Conference on Rewriting Techniques and Applications, Dijon, France, 1986.

- [Clark 79] D.W. Clark, "Measurements of Dynamic List Structures Use in Lisp", IEEE Transactions on Software Engineering SE-5(1), pp. 51-59. (Jan. 1979).
- [Dieu 68] J. Dieudonné, *Calcul infinitésimal*, Hermann, 1968.
- [Dufr 84] P. Dufresne, "Contribution algorithmique à l'inférence par règles de production", Thèse Université Paul Sabatier, Toulouse, 1984.
- [DNM 78] J. McDermott, A. Newell, J. Moore, "The efficiency of certain production system implementations", dans *Pattern-Directed Inference Systems* (Waterman et Hayes-Roth ed.) Academic Press, New York, 1978, pp. 155-176.
- [Fag 86] F. Fages, "On the proceduralization of rules in expert systems", First France-Japan Symposium on Artificial Intelligence, Programming of Future Generation Computers, Addison-Wesley, Eds. M. Nivat and K. Fuchi, Tokyo, Nov., 1986.
- [Fag 88] F. Fages, "Rulebased extension of programming languages", Proceedings of Les systèmes experts et leurs applications, Avignon, 1988.
- [Fla 85] P. Flajolet, "Mathematical methods in the analysis of algorithms and data structures", INRIA, Research Report 400, 1985. To appear in A Graduate Course in Computer Science, Computer Science Press, 1987.
- [Fla 87] P. Flajolet, "The symbolic operator method", *Mathematical methods in the analysis of algorithms and data structure*, L.N.C.S., Springer Verlag, to appear, 1987.
- [Forg 79] C. Forgy, "On the efficient implementation of production systems", PhD Thesis, Carnegie Mellon University, 1979.
- [Forg 81] C. Forgy, "OPS-V user's manual", Computer Science Department, Carnegie Mellon University, Pittsburgh, MA, 1981.
- [Forg 82] C. Forgy, "RETE, a fast algorithm for the many patterns many objects Match problem", *Artificial Intelligence* 19, 1982, pp. 17-37.
- [FS 86] P. Flajolet, R. Sedgewick, "Mathematical analysis of algorithms". Computer Science 504, Lecture Notes for Princeton University, 1986.
- [FSS] P. Flajolet, P. Sipala et J.M. Steyaert, "The analysis of tree compaction in symbolic manipulations", preprint.
- [FV 87] P. Flajolet, J. Vitter, "Average Case Analysis of Algorithms and Data Structures", dans *A Handbook of Theoretical Computer Science*, North-Holland, 1987.
- [GD 84] M. Ghallab, P. Dufresne, "Moteurs d'inférences pour systèmes de règles de production : techniques de compilation et d'interprétation", Colloque d'Intelligence Artificielle, Marseille, Oct. 1984, pp. 89-103.
- [GF 83] A. Gupta et C.L. Forgy, "Measurements on production systems", Carnegie Mellon University, Technical Report CMU-CS-83-167, 1983.
- [Gha 80] M. Ghallab, "New optimal decision tree for matching patterns in inference and planning system", 2nd Int. Meeting on Artificial Intelligence, Leningrad, Oct. 1980.
- [GP 84] D. Gardy, C. Puech, "On the size of projection: a generating function approach", *Information Systems*, Vol. 9, No 3/4, pp. 231-235, 1984.
- [GP 88] D. Gardy, C. Puech, "On the effect of join operations on relation sizes", to appear in *ACM Transactions On Database Systems*.
- [Gupt 84] A. Gupta, "Parallelism in production Systems : the sources and the expected Speed-up", Carnegie Mellon University Technical Report CMU-CS-84-169, 1984.
- [Henr 74] P. Henrici, *Applied and Computational complex Analysis*, Volumes 1-3, Wiley, New-York.

- [LCR 88] Laboratoire central de Recherches, *Xrete : manuel de référence*, Thomson-CSF, domaine de Corbeville 91401 Orsay Cedex France, 1988.
- [Mir 87] D.P. Miranker, "TREAT: A Better Match Algorithm for AI Production Systems", Proceedings of the 1987 National Conference on Artificial Intelligence, Seattle, Washington, 1987.
- [MM 78] A. Meier, J.W. Moon, "On the altitude of nodes in random trees", *Canadian Journal of Math* 30, 1978, pp. 997-1015.
- [Rosen 81] A.S. Rosenthal, "Note on the expected size of a join", *SIGMOD Record* 11(4), pp. 19-25, july 1981.
- [RS 89] M. Regnier, E. Simon, "Efficient evaluation of production rules in a DBMS", to appear, 1989.
- [SF 83] J.M. Steyaert, P. Flajolet, "Patterns and pattern matching in trees : an analysis", *Information and Control* 58, 1983, pp. 19-58.
- [SF 88] Schang T. and Fages F. "A Real-Time Expert System for On-Board Radar Identification" 55th Symposium AVP-AGARD on Software Engineering and its Applications to Avionics, 1988.
- [Stey 84] J.M. Steyaert, "Complexité et structures des algorithmes", Thèse d'Etat, Université de Paris 7, 1984.
- [Vien 86] G. Viennot, "La combinatoire bijective par l'exemple", Université de Bordeaux 1, 1986.
- [WGF 86] M.L. Wright, M.W. Green, G.Fiegl, P.F. Cross, "An Expert System for Real-Time Control", SRI International, in *IEEE Software*, March 1986.

Annexe

Nous reprenons ici l'exemple des trois règles du 2.1 dont le graphe de RETE est présenté en figure 2. Nous avons précisé au 3.1 les caractéristiques de cet exemple (séries génératrices, tailles moyennes ...). Nous présentons dans cette annexe sur le dessin du graphe les taux de filtrage en chacun des nœuds pour les modèles de liste et de multiensemble avec un seul domaine de variation de constante et une distribution uniforme. Les taux de filtrage sont notés en *italique* pour le modèle de liste et en **gras** pour le modèle de multiensemble.



(Pour des commodités d'écriture, on n'a écrit qu'une seule fois les taux de filtrage aux nœuds 4, 5, 6 et 7, 8, 9. On a en effet $\overline{taux}_4 = \overline{taux}_5 = \overline{taux}_6$ et $\overline{taux}_7 = \overline{taux}_8 = \overline{taux}_9$ dans les deux modèles).

On a obtenu le taux de filtrage au nœud 12 sachant que :

$$\overline{taux}((g \ X \ Y), \neg(f \ a0 \ X)) = \overline{taux}(g \ X \ Y) - \overline{taux}((g \ X \ Y), (f \ a0 \ X))$$

Remarque : Pour obtenir le nombre moyen de termes ou de l -uplets de termes filtrés en un nœud i , il faut multiplier le taux de filtrage par k^l avec k taille de la Base de Faits d'entrée et l longueur des uplets de termes produits en sortie de i .

